

IMECE2022-95229

INVESTIGATIONS INTO UNCERTAIN CONTROL CO-DESIGN IMPLEMENTATIONS FOR STOCHASTIC IN EXPECTATION AND WORST-CASE ROBUST

Saeed Azad

Postdoctoral Fellow

Department of Systems Engineering
Colorado State University
Fort Collins, CO 80523

Email: saeed.azad@colostate.edu

Daniel R. Herber*

Assistant Professor

Department of Systems Engineering
Colorado State University
Fort Collins, CO 80523

Email: daniel.herber@colostate.edu

ABSTRACT

As uncertainty considerations become increasingly important aspects of concurrent plant and control optimization, it is imperative to identify and compare the impact of uncertain control co-design (UCCD) formulations on their associated solutions. While previous work has developed the theory for various UCCD formulations, their implementation, along with an in-depth discussion of the structure of UCCD problems, implicit assumptions, method-dependent considerations, and practical insights, is currently missing from the literature. Therefore, in this study, we address some of these limitations by proposing two optimal control structures for UCCD problems that we refer to as the open-loop single-control (OLSC) and open-loop multiple-control (OLMC). Next, we implement the stochastic in expectation UCCD (SE-UCCD) and worst-case robust UCCD (WCR-UCCD) for a simplified strain-actuated solar array (SASA) case study. For the implementation of SE-UCCD, we use generalized Polynomial Chaos expansion and benchmark the results against Monte Carlo Simulation. Next, we solve a simple SASA WCR-UCCD through OLSC and OLMC structures. Insights from such implementations indicate that constructing, implementing, and solving a UCCD problem requires an in-depth understanding of the problem at hand, formulations, and solution strategies to best address the underlying co-design under uncertainty questions.

Keywords: control co-design; uncertainty; generalized polynomial chaos; worst-case robust; polytopic uncertainties

1 INTRODUCTION

In uncertain control co-design (UCCD) problems, uncertainty is a challenging and unavoidable aspect of modeling system behavior and realizing pragmatic design solutions. In our previous work, we offered a broad overview of uncertainties, their interpretation, and ways to integrate them into various UCCD formulations [1]. This resulted in several UCCD problems motivated by concepts from stochastic programming [2,3], robust optimization [4–6], and fuzzy programming [7–9]. However, an effective solution strategy is required to answer the design optimization question put forth by a novel UCCD formulation. In this article, we start to fill this gap by focusing on implementations for several unique and illustrative UCCD formulations with a focus on UCCD problems with open-loop control.

To this end, we first discuss the optimal, open-loop control structures under uncertainties. In this context, we use the term *control structure* to refer to various ways in which control trajectories can be structured in the UCCD problem in a physically-meaningful manner (and is different from CCD coordination strategies, such as simultaneous and nested [10–12]). The need for the discussion on control structures is motivated by the role that optimal control trajectories play in UCCD problems. We propose two structures: 1) *open-loop single-control* (OLSC) and 2) *open-loop multiple-control* (OLMC). These structures are related to concepts from worst-case robust [13–16] and stochastic [3,17,18] interpretations of uncertainty, respectively. Insights from our discussion on control structures, presented in Sec. 2, are expected to provide a more meaningful approach for meeting

*Corresponding author

problem requirements in the presence of uncertainties.

Then, a simple CCD problem is used as a case study (adopted from Refs. [19, 20]) and modified to include all time-independent, natural uncertainties. These uncertainties stem from plant optimization variables, initial states, and problem data and result in a UCCD problem that can be formulated like any of the forms discussed in Ref. [1]. The simple strain-actuated solar array (SASA) UCCD problem [19, 20] was selected due to its simplicity in both implementation and the interpretation of results. In this study, we develop and implement the simple SASA problem for two different UCCD formulations: stochastic in expectation UCCD and worst-case robust UCCD. Since the solution of UCCD problems with probabilistic uncertainties requires an appropriate uncertainty propagation (UP) method, we also introduce Monte Carlo Simulation (MCS) and the non-intrusive, collocation-type, generalized Polynomial Chaos (gPC). By avoiding problem-dependent complexities, this article allows for a straightforward comparison of such formulations—highlighting the impact of problem formulation and method-dependent decisions on the UCCD solution.

1.1 Deterministic Control Co-design

The nominal continuous-time, deterministic, all-at-once (AAO), simultaneous, CCD problem is formulated as [10, 21]:

$$\text{minimize: } o = \int_{t_0}^{t_f} \ell(t, \mathbf{u}, \boldsymbol{\xi}, \mathbf{p}, \mathbf{d}) dt + m(\mathbf{p}, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f, \mathbf{d}) \quad (1a)$$

$$\text{subject to: } \mathbf{g}(t, \mathbf{u}, \boldsymbol{\xi}, \mathbf{p}, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f, \mathbf{d}) \leq \mathbf{0} \quad (1b)$$

$$\mathbf{h}(t, \mathbf{u}, \boldsymbol{\xi}, \mathbf{p}, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f, \mathbf{d}) = \mathbf{0} \quad (1c)$$

$$\dot{\boldsymbol{\xi}} - \mathbf{f}(t, \mathbf{u}, \boldsymbol{\xi}, \mathbf{p}, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f, \mathbf{d}) = \mathbf{0} \quad (1d)$$

$$\text{where: } \boldsymbol{\xi}(t_0) = \boldsymbol{\xi}_0, \boldsymbol{\xi}(t_f) = \boldsymbol{\xi}_f, \mathbf{u}(t) = \mathbf{u}, \boldsymbol{\xi}(t) = \boldsymbol{\xi} \quad (1e)$$

$$\mathbf{d}(t) = \mathbf{d}$$

where $t \in [t_0, t_f]$ is the time horizon, $\{\mathbf{u}, \boldsymbol{\xi}, \mathbf{p}\}$ are the collection of optimization variables including the open-loop control trajectories $\mathbf{u}(t) \in \mathbb{R}^{n_u}$, state trajectories $\boldsymbol{\xi}(t) \in \mathbb{R}^{n_s}$, and the vector of time-independent optimization variables $\mathbf{p} \in \mathbb{R}^{n_p}$, respectively. Note that \mathbf{p} may entail plant optimization variables \mathbf{p}_p and/or time-independent control optimization variables \mathbf{p}_c [19, 22] (i.e., gains), such that $\mathbf{p} = [\mathbf{p}_p, \mathbf{p}_c]$. The objective function $o(\cdot)$ is composed of the Lagrange term $\ell(\cdot)$ and the Mayer term $m(\cdot)$. The vectors of inequality and equality constraints are described by $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$, respectively. The transition or state derivative function $\mathbf{f}(\cdot)$ describes the evolution of the system states through time in terms of a set of ordinary differential equations (ODEs). All of the time-dependent or time-independent data associated with the problem formulation, such as problem constants, environmental signals, initial/final times, etc., is represented through $\mathbf{d} \in \mathbb{R}^{n_d}$.

In the remainder of this article, we assume that constraints associated with the initial and final conditions $\{\boldsymbol{\xi}_0, \boldsymbol{\xi}_f\}$ are already included in $\mathbf{h}(\cdot)$ or $\mathbf{g}(\cdot)$. We will often drop the explicit depen-

dence on t from time-dependent quantities such as control and state trajectories, as well as the problem data. For more details on deterministic CCD, the readers are referred to Refs. [10, 11].

1.2 Uncertain CCD Problem Formulation

Now, we introduce an AAO, continuous-time, simultaneous UCCD formulation in the probability space as introduced in Ref. [1]:

$$\text{minimize: } \mathbb{E}[\bar{o}(t, \tilde{\mathbf{u}}, \tilde{\boldsymbol{\xi}}, \tilde{\mathbf{p}}, \tilde{\mathbf{d}})] \quad (2a)$$

$$\text{subject to: } \mathbb{E}[\tilde{\mathbf{g}}(t, \tilde{\mathbf{u}}, \tilde{\boldsymbol{\xi}}, \tilde{\mathbf{p}}, \tilde{\mathbf{d}})] \leq \mathbf{0} \quad (2b)$$

$$\mathbf{h}(t, \tilde{\mathbf{u}}, \tilde{\boldsymbol{\xi}}, \tilde{\mathbf{p}}, \tilde{\mathbf{d}}) = \mathbf{0} \quad (2c)$$

$$\dot{\tilde{\boldsymbol{\xi}}}(t) - \mathbf{f}(t, \tilde{\mathbf{u}}, \tilde{\boldsymbol{\xi}}, \tilde{\mathbf{p}}, \tilde{\mathbf{d}}) = \mathbf{0} \quad (2d)$$

$$\text{where: } \tilde{\mathbf{u}}(t) = \tilde{\mathbf{u}}, \tilde{\boldsymbol{\xi}}(t) = \tilde{\boldsymbol{\xi}}, \tilde{\mathbf{d}}(t) = \tilde{\mathbf{d}} \quad (2e)$$

where the expectation of $\bar{o}(\cdot)$, which is a function composition of $o(\cdot)$, is optimized over the set of optimization variables $(\tilde{\mathbf{u}}, \tilde{\boldsymbol{\xi}}, \tilde{\mathbf{p}})$, and is subject to the expectation of $\tilde{\mathbf{g}}(\cdot)$ (i.e., a function composition of $\mathbf{g}(\cdot)$), analysis-type equality constraints $\mathbf{h}(\cdot)$, and uncertain dynamic system equality constraints in Eq. (2d). Note that $\mathbb{E}[\bar{o}(\cdot)]$ and $\mathbb{E}[\tilde{\mathbf{g}}(\cdot)]$ may be any of the variations that are discussed in Ref. [1] (such as the nominal, worst-case, expected value, etc.). This formulation includes the vector of uncertain control processes $\tilde{\mathbf{u}}(t)$, uncertain state processes $\tilde{\boldsymbol{\xi}}(t)$, time-independent uncertain optimization variables $\tilde{\mathbf{p}}$, and time-dependent/time-independent uncertain problem data $\tilde{\mathbf{d}}$.

This formulation is infinite-dimensional in time and uncertainty dimensions. We can draw an analogy between the infinite-dimensional time vector and the infinite-dimensional uncertainty vector. To transcribe Eq. (2) in time, numerical methods such as direct transcription have been implemented [11, 23–26]. Similarly, different (UP) techniques, such as MCS and gPC, or special interpretations, such as worst-case robust, have been proposed to parameterize the uncertain dimensions. Note that describing optimization variables $(\tilde{\mathbf{u}}, \tilde{\boldsymbol{\xi}}, \tilde{\mathbf{p}})$ in the uncertain space is to avoid introducing any unnecessary structure into these variables at this point. This description certainly does not imply that the designer has full control over all of the uncertainties; rather, it implies that the decision variables may entail elements associated with uncertainties, such as mean values, standard deviation, or even entire distributions.

Through the appropriate selection of the objective function and constraints, specialized formulations such as stochastic in expectation (SE-UCCD), stochastic chance-constrained (SCC-UCCD), worst-case robust (WCR-UCCD), probabilistic robust (PR-UCCD), fuzzy expected value (FE-UCCD), and possibilistic chance-constrained (PCC-UCCD) can be derived. All of these formulations are discussed in detail in Ref. [1].

The remainder of this article is organized in the following manner: Sec. 2 describes some considerations in implementing UCCD problems; UP methods such as MCS and gPC are intro-

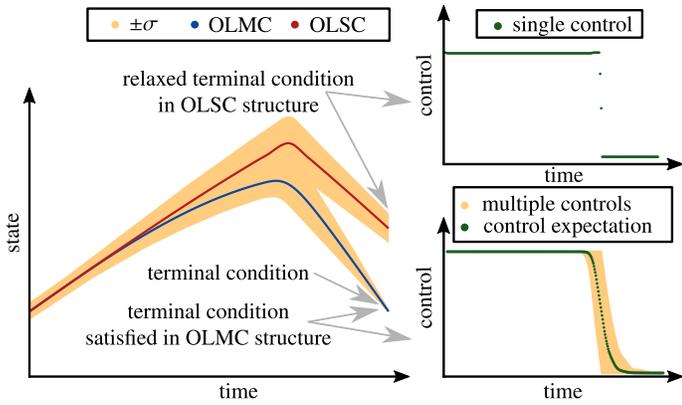


FIGURE 1: Illustration of OLS and OLMC structures and their effects on states of an arbitrary boundary-value UCCD problem.

duced in Sec. 3; SE-UCCD and WCR-UCCD formulations associated with the simple SASA case study are described in Sec. 4; and Sec. 5 presents results and discussion from such implementations. Finally, Sec. 6 presents the conclusions.

2 UCCD IMPLEMENTATION

In this section, we start by discussing the open-loop, optimal control structures under uncertainties in UCCD problems. These structures attempt to answer inherently different questions and have been used in separate studies in the literature [16, 27–35]. While these structures have a significant impact on the problem implementation and its associated solution, they have not been collectively discussed in the literature to the best of the authors’ knowledge. This section also highlights some subtle implementation considerations and extends the discussion to the effectiveness of various coordination strategies, such as simultaneous, nested, and direct single shooting (DSS), for general UCCD problems [10–12, 26].

2.1 Control Structures under Uncertainties

To respond to the presence of uncertainties and manage their impact on a UCCD problem, one may use an OLS or an OLMC structure. OLS is structured to find a single control command, while OLMC elicits a range of optimal control responses based on the realization of uncertainties. Here, we discuss these organizational structures in detail.

2.1.1 Open-Loop Single-Control (OLS). In this structuring, the goal is to find a single control command that is optimal with respect to some criteria, such as the expectation of the objective function or the worst-case uncertainty realization. In that sense, OLS is closely related to concepts from robust control theory because the single robust control command is to perform well under a range of circumstances. As an example, OLS

can be used to generate an open-loop optimal control trajectory for motion planning of a robot, which is then followed using a tracking controller. While OLS is particularly suitable for generating open-loop trajectories to be used in trajectory tracking applications, there are some inherent limitations to this structure.

For instance, consider a general UCCD problem with initial and terminal boundary conditions. This class of problems, which is referred to as boundary-valued UCCD, has an extensive application in various fields of engineering. As an example, a boundary-valued UCCD problem might attempt to bring a vehicle to rest, an aircraft back to a depot, the energy of a system to zero, or the temperature in a heat-transfer system to a specific value—all in the presence of uncertainties. For simplicity, assume that uncertainties in our boundary-valued UCCD problem originate only from uncertain plant optimization variables. In this case, it becomes immediately evident that the OLS structure cannot satisfy all of the prescribed initial and terminal boundary conditions in the presence of uncertainties. In other words, there’s no single control that can satisfy the prescribed boundary conditions for all of the realizations of plant uncertainty because the OLS problem is *over-constrained*.

This issue has been dealt with in two different ways in the literature: (1) relaxing the prescribed terminal boundary conditions [27, 28], or (2) minimizing the variance of the terminal state in a multi-objective optimization problem [29, 30]. The premise for both of these approaches is the assumption that terminal conditions are of Type II equality constraints (i.e., the strict satisfaction of such constraints under uncertainty cannot be guaranteed) [1, 36, 37]. These remedies enable a solution to the OLS-UCCD problem, but they do not enforce the terminal boundary conditions. As a result, the relaxed UCCD problem is an inherently different problem to solve. This issue is illustrated in Fig. 1, where the boundary value is relaxed for the OLS structure to enable a solution. In addition, in many real-world applications, relaxing the boundary conditions is not practically viable. For example, in the UCCD problem of a military aircraft, the vehicle must reach the target or depot despite all uncertainties. Therefore, relaxing the terminal condition has the potential to change the meaning and interpretation of the problem, and thus, care must be taken at the time of implementation. Despite these limitations, the OLS uncertainty organization remains a valuable tool in constructing open-loop trajectories for a wide range of reference tracking applications.

In terms of implementation, OLS generally requires the relaxation of some or all of the boundary conditions. For a general UCCD problem, a nested coordination strategy may be leveraged to deal with plant uncertainty only in the outer-loop optimization problem. However, the inner-loop optimal control subproblem will remain coupled through control trajectories—resulting in a problem that is too large to solve unless a more suitable UP method, such as the most-probable-point (MPP), is used to limit the size of the inner-loop problem [27, 38]. One remedy for deal-

ing with this issue is to use a DSS method [25, 26, 39] because it has the advantage of effectively separating control variables from dynamic equations, resulting in sets of dynamic equations that can be solved efficiently by taking advantage of parallel computation. For this reason, all of the implementations of OLSC-UCCD problems in this study use a DSS approach.

2.1.2 Open-Loop Multiple-Control (OLMC). OLMC is based on the idea that every realization of uncertainty should elicit a distinct optimal control response from the UCCD problem and is implemented in Refs. [31, 33, 40–43]. Even when control trajectories are not the source of uncertainty or when the control loop is not closed, this approach results in an aggregate of optimal control solutions that provide additional insights into the limits of the system performance. As an example, OLMC can be used to discover the upper system performance limits at the early-stage design of a vehicle active-suspension system, such as the maximum actuator force, maximum sprung mass acceleration, etc.

One way to view OLMC is to consider the closed-form solution to the optimal supervisory control for an arbitrary UCCD problem. Since the solution can be written as a function of other problem elements, such as plant optimization variables or problem data, then uncertainties in these quantities elicit a range of optimal control responses. In addition, OLMC enables investigations into the upper limits of system’s performance in early-stages of design. Therefore, the collection of these control trajectories can then be used to provide insights into defining the control architecture.

In addition, OLMC enables a more balanced UCCD formulation by allowing the designer to utilize and exploit the whole decision space in response to uncertainties. As an example, consider a system in which uncertainties stem from plant optimization variables. While it is possible to shift the plant design in response to uncertainties in order to achieve reliability or robustness (using methods based on reliability-based CCD [27, 44], or robust CCD [45], respectively), it might be more cost-effective to leverage the control limits to achieve such criteria. Therefore, OLMC is more suitable for early-stage design where plant and control spaces are explored, not only for performance optimality but also for reliability, robustness, or any other risk measures.

The idea of OLMC, introduced here, has some key differences from various control trajectories that are generated in a closed-loop controller. When the control loop is closed, different realizations of uncertainties result in different trajectories (which is also the case with OLMC). However, not all of these trajectories result in optimal behavior. This is in contrast with the OLMC structure discussed in this section, which seeks to generate control trajectories that are optimal for every realization of uncertainties. Formally connecting these optimal MC trajectories and more effective and implementable closed-loop controllers in later stages of the design process is important future work.

The implementation of OLMC structure in the simultaneous coordination strategy for any general UCCD problem with at least plant uncertainty¹ requires expanding the number of control variables to match the number of samples or function evaluation points. This result in a prohibitively large UCCD problem. However, using a nested coordination strategy, the inner-loop optimal control subproblems associated with sample points become completely independent and can be solved more efficiently. In other words, control variables along with their associated dynamic equations are decoupled and can be solved in parallel within the nested coordination strategy. For this reason, all of the instances of OLMC-UCCD implementations for stochastic formulations in this study use a nested coordination strategy.

2.2 Simple Bounds in UCCD

While there is limited theoretical distinction between simple bounds and general inequality/path constraints, suitable treatment of such boundary constraints in the presence of uncertainties is occasionally overlooked in the literature. This distinction is especially problematic because if the solution to the UCCD problem lies on or near the boundary, it might become practically infeasible. To illustrate this further, consider a deterministic CCD problem in which the plant optimization variable p_p is bounded $p_{p_{min}} \leq p \leq p_{p_{max}}$. When imperfect manufacturing processes, which tend to have a normal distribution, transform p_p into an uncertain variable \tilde{p}_p , the associated bounds on \tilde{p}_p need to be modified to reflect the true decision space. Since, in this special case, \tilde{p}_p has a normal distribution, a natural way to adjust its bounds is to include a constraint shift index k_s such that:

$$p_{p_{min}} + k_s \sigma_p \leq \tilde{p}_p \leq p_{p_{max}} - k_s \sigma_p \quad (3)$$

where k_s is chosen by the designer to reflect the risks associated with these simple bounds. As an example, when $k_s = 3$ and \tilde{p}_p has a normal distribution, then the simple bound is satisfied with the probability of 99.865%. However, this approach does not always offer this probabilistic interpretation. The concept of constraint shift index, borrowed from robust design optimization [46], is implemented in multiple UCCD studies [27, 28, 38, 45]. Using this approach, we can ensure that the UCCD problem accounts for this specific type of reduced decision space that results from the underlying uncertainties, and thus, any boundary solution will remain feasible and meaningful.

3 UNCERTAINTY PROPAGATION METHODS

Uncertainty propagation (UP) is a term applied to a family of methods that quantify uncertainties in a system’s response and

¹With plant uncertainty, the UP method needs to be constructed inside the optimization problem and evaluated at every iteration of the optimization solver. This results in a problem structure that cannot be solved sequentially through a simultaneous coordination strategy.

is a critical step in UCCD solution strategies when uncertainties are represented probabilistically. UP methods have been developed for different needs in various research communities, and thus, it is important to understand their utility, merits, and limitations in the context of UCCD. In a broad classification, UP may be performed in a forward or inverse manner [47, 48]. In forward UP, uncertainties from inputs are propagated through the system model, resulting in predictions of system behavior in the presence of uncertainties. Inverse UP, on the other hand, uses experimental measurements or observations to estimate discrepancies between the model and the actual system. Parameter calibration and bias correction are then performed in order to update the model based on such observations. While inverse UP is an important step in developing accurate models, it is often the case that actual observations and experimentation are not viable options in early-stage design. In addition, while a concurrent forward and inverse UP algorithm has the potential to improve the dynamic model and hence, the interpretation of UCCD solutions, this gain comes at the cost of a high computational burden. For these reasons, this article only considers methods for forward propagation.

Forward UP methods can be divided into probabilistic and non-probabilistic approaches. Probabilistic methods provide an estimate of the likelihood of an event taking place. This class includes sampling methods such as MCS [49], local expansion methods such as first-order second-moment (FOSM) [45] and perturbation [50–52], functional expansion methods such as gPC [53, 54] and Karhunen–Loève expansion [55], MPP-based methods such as first-order reliability method (FORM) [27, 56, 57] and second-order reliability method (SORM) [58], and numerical integration methods such as full-factorial numerical integration [59, 60]. Non-probabilistic methods, such as interval analysis, or methods based on fuzzy programming, on the other hand, do not provide this information, and their usage is generally limited to conditions where sufficient data is not available [61]. In addition, the time-evolution of the joint probability distribution function (PDF) in an uncertain system can be estimated directly using direct quadrature method of moments [62], direct evolution through Fokker-Planck-Kolmogorov equation [63], and the Riccati equation (for linear Gaussian systems) [64]. In this section, we limit our discussion to MCS and gPC and discuss their utility and implementation challenges for UCCD problems.

3.1 Monte Carlo Simulation (MCS)

UP methods that are based on sampling provide an intuitive way of propagating and quantifying uncertainties in many engineering problems. These methods generally require a lot of empirical information. For example, not only the mean value and variance but also the complete probability distribution of uncertain variables must be known for an effective sampling of uncertainties [65]. These samples can be generated in two ways. The generated samples can be stored in computer memory and called for

appropriate function evaluations, or the same sample can be generated using a common seed number in a pseudorandom number generator [66].

The sampling methods encompass a wide range of approaches, among which MCS is the most well-known. In MCS, random samples of uncertain variables are generated from their joint PDF, and the model is evaluated repeatedly for these samples. Therefore, at the root of methods such as MCS is the availability of distributional information for *uncertain basic quantities*. In this context, uncertain basic quantities are variables described as $\tilde{\mathbf{x}} \in \mathbb{R}^{n_x}$ that have a known PDF whose uncertainty will be propagated into the system. As an example, when the uncertainty in an arbitrary UCCD problem stems only from plant optimization variables, then $n_x = n_p$. Using $\tilde{\mathbf{x}}$ to describe the source of uncertainties will allow us to simplify some of the notations in the remainder of the article. An unbiased estimator for the mean value of the objective function using MCS can be calculated as:

$$\hat{\delta}_\mu = \frac{1}{N_{\text{mcs}}} \sum_{j=1}^{N_{\text{mcs}}} o(t, \mathbf{u}_j, \boldsymbol{\xi}_j, \mathbf{p}_j, \mathbf{d}_j) \quad (4)$$

where N_{mcs} is the number of samples. The variance can be estimated as:

$$\hat{\delta}_\sigma^2 = \frac{1}{N_{\text{mcs}} - 1} \sum_{j=1}^{N_{\text{mcs}}} (o(t, \mathbf{u}_j, \boldsymbol{\xi}_j, \mathbf{p}_j, \mathbf{d}_j) - \hat{\delta}_\mu)^2 \quad (5)$$

To estimate the probability of failure for an arbitrary inequality constraint g_i (i.e., $\mathbb{P}[g_i(\cdot) > 0]$), an unbiased estimator is calculated as:

$$\hat{\mathbb{P}}_{f,i} = \frac{1}{N_{\text{mcs}}} \sum_{j=1}^{N_{\text{mcs}}} \mathbb{I}(t, \mathbf{u}_j, \boldsymbol{\xi}_j, \mathbf{p}_j, \mathbf{d}_j) \quad (6)$$

where $\mathbb{I}(\cdot)$ is an indicator function defined as:

$$\mathbb{I}(t, \mathbf{u}_j, \boldsymbol{\xi}_j, \mathbf{p}_j, \mathbf{d}_j) = \begin{cases} 0 & \text{if } g_i(\cdot) \leq 0 \\ 1 & \text{if } g_i(\cdot) > 0 \end{cases} \quad (7)$$

Equations (4)–(7) can now be used in Eq. (2) to produce any of the standard or specialized formulations based on stochastic and probabilistic robust formulations [1].

MCS is flexible, easy to implement, and capable of offering high solution accuracy. Therefore, it is often used to benchmark newly-developed methods. However, a prohibitively large number of samples are required to estimate rare events accurately (i.e., event with $\mathbb{P} \ll 1$). The convergence rate of the expected value in MCS is $\mathcal{O}(1/\sqrt{N_{\text{mcs}}})$ [49, 54]. Interestingly, this convergence rate does not depend on the dimension n_x , which is an advantage when compared to some of the other UP methods.

3.2 Generalized Polynomial Chaos (gPC)

The following presentation of gPC is primarily based on Ref. [67]. In gPC, stochastic variables and processes are parameterized and represented using orthogonal polynomials. Suppose that $\tilde{\mathbf{x}}$ is a random vector with mutually independent components

and distribution $F_{\tilde{\mathbf{x}}}(\mathbf{x}) = \mathbb{P}(\tilde{x}_1 \leq x_1, \dots, \tilde{x}_{n_x} \leq x_{n_x})$. If the random elements in $\tilde{\mathbf{x}}$ are not independent, then a Karhunen–Loève decomposition [68] or a Rosenblatt transformation [69] should first be applied. The mutual independence implies that $F_{\tilde{\mathbf{x}}}(\mathbf{x}) = \prod_{i=1}^{n_x} F_{\tilde{x}_i}(x_i)$. The univariate gPC basis functions of degree up to r_i are denoted by $\{\phi_k(\tilde{x}_i)\}_{k=0}^{r_i}$ and, for continuous \tilde{x}_i , satisfy the following orthogonality condition:

$$\mathbb{E}[\phi_m(\tilde{x}_i)\phi_n(\tilde{x}_i)] = \int \phi_m(\mathbf{x})\phi_n(\mathbf{x})dF_{\tilde{x}_i}(\mathbf{x}) = \delta_{mn}\gamma_m \quad (8)$$

$$\text{where } 0 \leq m, n \leq r_i, \quad \gamma_m = \mathbb{E}[\phi_m^2(\tilde{x}_i)]$$

where δ is the Kronecker delta function and γ is a normalization factor. The set of univariate orthogonal basis functions are simply obtained based on the probability distribution of $\tilde{\mathbf{x}}$. As an example, when \tilde{x}_i has a Gaussian distribution, the set of orthogonal basis functions are selected from Hermite orthogonal polynomials, that is $\{\phi_k(\tilde{x}_i)\}_{k=0}^{r_i} = \{H_k(\tilde{x}_i)\}_{k=0}^{r_i} = \{H_0(\tilde{x}_i), H_1(\tilde{x}_i), \dots, H_{r_i}(\tilde{x}_i)\}$, where $H(\cdot)$ refers to Hermite polynomial basis. Note that if $\tilde{\mathbf{x}}$ does not follow the random distributions in the Askey family, as described in Table 1, then data-driven polynomial chaos can be used [70, 71].

A tensor product of elements in $\{\phi_k(\tilde{x}_i)\}_{k=0}^{r_i}$, for $i = 1, \dots, n_x$ can then be used to construct the n_x -variate gPC basis functions $\Phi_m(\tilde{\mathbf{x}})$. The resulting polynomials space can be defined as:

$$\left\{ \Phi_m(\tilde{\mathbf{x}}) \right\}_{m=0}^{M-1} = \otimes_{|\mathbf{k}| \leq PC} \left\{ \prod_{i=1}^{n_x} \phi_k(\tilde{x}_i) \right\} \quad (9)$$

where \otimes indicates the tensor product. If we keep the highest polynomial order for up to PC in each direction, then $|\mathbf{k}| = \max_{1 \leq j \leq n_x} k_j$ and the resulting polynomials have the dimension of $M = \prod_{i=1}^{n_x} (PC + 1)$. Alternatively, a subset of basis elements with a total degree of up to PC can be selected. In this case, the n_x -variate gPC is defined using a multi-index $\mathbf{k} = (k_1, \dots, k_{n_x}) \in \mathbb{N}_0^{n_x}$, where $\mathbb{N}_0^{n_x}$ is the set of n_x -dimensional natural numbers with zero, and \mathbf{k} is a multi-index with $|\mathbf{k}| = k_1 + \dots + k_{n_x}$. The resulting polynomials have the dimension of M :

$$M = \binom{PC + n_x}{PC} \quad (10)$$

The n_x -variate gPC basis functions satisfy the following orthogonality condition:

$$\mathbb{E}[\Phi_i(\tilde{\mathbf{x}})\Phi_j(\tilde{\mathbf{x}})] = \int \Phi_i(\mathbf{x})\Phi_j(\mathbf{x})dF_{\tilde{\mathbf{x}}}(\mathbf{x}) = \gamma_i\delta_{ij} \quad (11)$$

where γ is the normalization factors and δ is the n_x -variate Kronecker delta functions. This orthogonality allows us to use suitable polynomials as basis functions to approximate a function of random variables $\tilde{\mathbf{x}}$ [67]. Now, any general second-order variable or process (i.e., a process with finite variance) $\tilde{y}(t, \tilde{\mathbf{x}})$ can be expressed by polynomial chaos of PC degree as:

$$\tilde{y}(t, \tilde{\mathbf{x}}) \approx y_{PC}(t, \tilde{\mathbf{x}}) = \sum_{m=0}^{M-1} \hat{y}_m(t)\Phi_m(\tilde{\mathbf{x}}) \quad (12)$$

TABLE 1: Correspondence between the type of gPC and their underlying random variables [67].

| | Distribution | gPC polynomial | Support |
|------------|-------------------|----------------|------------------------|
| Continuous | Gaussian | Hermite | $(-\infty, \infty)$ |
| | Gamma | Laguerre | $[0, \infty)$ |
| | Beta | Jacobi | $[a, b]$ |
| | Uniform | Legendre | $[a, b]$ |
| Discrete | Poisson | Charlier | $\{0, 1, \dots\}$ |
| | Binomial | Krawtchouk | $\{0, 1, \dots, N_d\}$ |
| | Negative binomial | Meixner | $\{0, 1, \dots\}$ |
| | Hypergeometric | Hahn | $\{0, 1, \dots, N_d\}$ |

where $y_{PC}(\cdot)$ is the PC th-degree gPC approximation of $\tilde{y}(\cdot)$, and $\hat{y}_i(t)$ are unknown coefficients. Note that in the context of UCCD problems, any uncertain problem element such as uncertain control or state trajectories, plant optimization variables, problem data, objective function, dynamic state equations, or constraints can be approximated using this approach.

From here, we can use either a Galerkin or a collocation formulation of gPC to calculate the unknown coefficients $\hat{y}_m(t)$ in Eq. (12). The Galerkin formulation, which is an intrusive UP method, uses a Galerkin projection on the basis functions to approximate uncertain quantities. For stochastic differential equations, the Galerkin projection results in an expanded deterministic set of coupled ODEs. Since model equations or the source code need to be modified for this approach, it is referred to as an intrusive UP method [72]. Solving the resulting coupled ODEs becomes increasingly difficult as the number of states increases. The development and application of the Galerkin type of gPC can be found in Refs. [29, 30, 67, 70].

In the collocation formulation of gPC, the m th unknown coefficient, $\hat{y}_m(t)$ can be obtained using Eq. (11):

$$\hat{y}_m(t) = \mathbb{E}[y(t, \mathbf{x})\Phi_j(\tilde{\mathbf{x}})] = \int_{\Gamma} y(t, \mathbf{x})\Phi_j(\mathbf{x})dF_{\tilde{\mathbf{x}}}(\mathbf{x}) \quad (13)$$

where Γ is the finite domain of the distribution function, constructed from the product of independent finite domains associated with each uncertain dimension, i.e., $\Gamma = \prod_{i=1}^{n_x} \Gamma_i$. The evaluation of this integral requires a quadrature rule, and thus, a set of collocation nodes, along with their associated quadrature weights must be selected according to the corresponding polynomial representation in Table 1. We define q_i user-defined collocation nodes in each i dimension of uncertain quantities. The full tensor product of these user-defined nodes can then be used to generate an n_x -dimensional grid of Q collection nodes and quadrature weights α_w , such that $\{\mathbf{x}_j, \alpha_w\}_{j=1}^Q$, where $Q = \otimes\{q_1, \dots, q_{n_x}\}$. Note that a sparse grid can also be generated through methods such as the Smolyak algorithm [73] to reduce the total number of tensor products, and thus the computational expense, especially in problems with higher uncertain dimensions, while maintaining

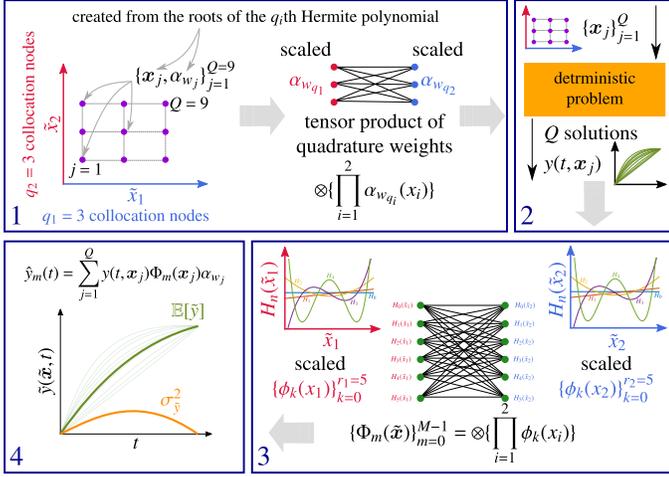


FIGURE 2: Illustration of steps involved in constructing gPC basis functions for a two-dimensional uncertain problem.

the accuracy of integration. Using the quadrature rule, Eq. (13) can be written as:

$$\hat{y}_m(t) = \sum_{j=1}^Q y(t, \mathbf{x}_j) \Phi_m(\mathbf{x}_j) \alpha_{w_j} \quad (14)$$

where \mathbf{x}_j is the j th collocation node and α_{w_j} is the associated quadrature weight. Note that now, $y(t, \mathbf{x}_j)$ is the deterministic solution to the j th sample of random vector. When an accurate gPC for an arbitrary random function $\tilde{y}(t, \tilde{\mathbf{x}})$ is constructed, the function can be analytically represented as a function of $\tilde{\mathbf{x}}$. Therefore, all statistical information of $\tilde{y}(\cdot)$ can be estimated with little computational effort [67]. Statistics of $\tilde{y}(t, \tilde{\mathbf{x}})$ are then calculated as:

$$\begin{aligned} \mathbb{E}[\tilde{y}(t, \tilde{\mathbf{x}})] &\approx \mathbb{E}[\tilde{y}_{PC}(t, \tilde{\mathbf{x}})] = \hat{y}_1(t) \\ \sigma_{\tilde{y}(t, \tilde{\mathbf{x}})}^2 &\approx \sigma_{\tilde{y}_{PC}(t, \tilde{\mathbf{x}})}^2 = \sum_{m=2}^M \hat{y}_m^2(t) \end{aligned} \quad (15)$$

Figure 2 illustrates all of the steps involved in constructing gPC basis functions for an arbitrary, two-dimensional uncertain problem. In this figure, x_1 and x_2 have a Gaussian distribution; this is why a Hermite polynomial function is used to obtain orthogonal basis functions. The collocation nodes and quadrature weights associated with each uncertain dimension are obtained by finding the roots of the q_i th Hermite polynomials and are appropriately scaled according to Refs. [31, 67]. The source code for the implementation of gPC in MATLAB is provided in Ref. [74].

4 SIMPLE SASA UCCD FORMULATIONS

In this section, we construct SE-UCCD and WCR-UCCD formulations to investigate the integrated UCCD solution of the simple SASA case study as a means to explore UCCD more broadly.

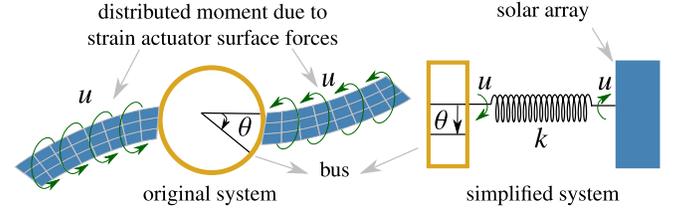


FIGURE 3: Illustration of the original and simplified strain-actuated solar array systems.

Borrowed from Refs. [19, 20], this problem describes a simplified version of a strain-actuated solar array (SASA) system for spacecraft pointing control and jitter reduction. As shown in Fig. 3, this system uses distributed actuators to strain the solar arrays, which results in reactive forces that can be used to control the spacecraft body. The objective is to maximize the displacement of the spacecraft's body at the final time. The deterministic simple SASA problem is formulated as:

$$\text{minimize: } -\xi_1(t_f) \quad (16a)$$

$$\text{subject to: } u - u_{max} \leq 0 \quad (16b)$$

$$u_{min} - u \leq 0 \quad (16c)$$

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (16d)$$

$$\xi(t_0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \xi_2(t_f) = 0 \quad (16e)$$

$$\text{where: } u(t) = u, \xi(t) = \xi \quad (16f)$$

In this problem, u is the open-loop control moment applied to the solar array, ξ_1 and ξ_2 are state variables (associated with the relative displacement and velocity), and k is the plant optimization variable associated with the stiffness of the solar array. The inertia ratio between the solar array and the bus is described by J , and considered problem data. The number of optimization variables is $n_u = 1$, $n_s = 2$, and $n_p = 1$. Note that u is bounded by u_{min} and u_{max} , and the initial and terminal boundary conditions are imposed through Eq. (16e).

When uncertainties are present, depending on the availability of distributional information, the simple SASA UCCD problem may be formulated like any of the specialized forms discussed in detail in Ref. [1]. In this article, we assume that uncertainties originate from plant optimization variable \tilde{k} , uncertain problem data \tilde{J} , and uncertain initial boundary condition for the second state variable $\tilde{\xi}_{2,t_0}$. Generally, time-dependent disturbances are also present in the dynamic model and may require specific treatment before they can be efficiently used in gPC. For this reason, this study only focuses on time-independent uncertainties. Next, we assume that uncertainties are represented in a probabilistic and deterministic manner, which enables the formulation of the UCCD problem in three distinct ways: SE-UCCD,

PR-UCCD, and WCR-UCCD—all of which can be derived from the universal UCCD formulation in Eq. (2).

For the simple SASA problem, it is important to bring the vehicle to rest at the final time. This condition may be necessary for accurate positioning, safety, and functionality of the system. Therefore, the simple SASA problem readily lends itself to the OLMC structure. In addition, since the satisfaction of the terminal boundary condition is guaranteed through OLMC structure, the standard deviation associated with the objective function is zero. Therefore, the probabilistic robust implementation practically reduces to stochastic in expectation. Thus, in the first step, this article implements and solves the OLMC stochastic in expectation UCCD problem using MCS and gPC. For WCR-UCCD formulation, it is more natural to use an OLSC structure. This is mainly due to difficulties associated with effectively sampling the uncertainty space when distributional information of uncertainties is lacking. However, due to the assumption of polytopic uncertainties, as well as the problem-specific requirement of bringing the vehicle to rest, we also investigate the OLMC implementation of this formulation.

In the following sections, we assume that all of these uncertain quantities have a Gaussian distribution with known standard deviations of σ_k , σ_J , and $\sigma_{\xi_{2,t_0}}$, respectively. Note, however, that other distributions according to Table 1 may also be used for gPC. In addition, for the deterministic representation of uncertainties, we assume that uncertainties belong to deterministic sets, constructed within the range of $\pm k_s \sigma$ of the mean values of uncertain quantities, where $k_s = 3$. Since the standard deviation of uncertainties has the potential to be controlled and reduced through further investments, we also investigate the WCR solution as a function of the standard deviation, using a scaling factor s_f . In addition, since the size and geometry of the uncertainty set, selected by the designer, has the potential to result in overly conservative or risky solutions, we also investigate the solution as a function of k_s .

4.1 Stochastic in Expectation (SE-UCCD)

When distributional information of uncertainties is available, the deterministic CCD problem may be formulated as SE-UCCD. In this formulation, the objective function and inequality constraints are modeled in terms of their expected values. This points to the *risk-neutral nature* of this formulation because no measures are taken to reduce the risks associated with constraints violation. The OLMC structure solves the problem subject to all boundary conditions by eliciting a distinct control response for every realization of uncertainties. The simultaneous SE-UCCD problem is formulated as:

$$\underset{u, \xi, \mu_k}{\text{minimize:}} \quad -\mathbb{E}[\tilde{\xi}_1(t_f)] \quad (17a)$$

$$\text{subject to:} \quad \text{Eqs. (16b)-(16c)} \quad (17b)$$

$$k_s \sigma_k - \mu_k \leq 0 \quad (17c)$$

$$\begin{bmatrix} \dot{\tilde{\xi}}_1 \\ \dot{\tilde{\xi}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\tilde{k} & 0 \end{bmatrix} \begin{bmatrix} \tilde{\xi}_1 \\ \tilde{\xi}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\tilde{J}} \end{bmatrix} u \quad (\text{a.s.}) \quad (17d)$$

$$\tilde{\xi}(t_0) = \begin{bmatrix} 0 \\ \mathcal{N}(\mu_{\xi_{2,t_0}}, \sigma_{\xi_{2,t_0}}) \end{bmatrix} \quad (17e)$$

$$\tilde{\xi}_2(t_f) = 0 \quad (\text{if OLMC}) \quad (17f)$$

$$\text{where: } \tilde{k} = \mathcal{N}(\mu_k, \sigma_k), \quad \tilde{J} = \mathcal{N}(\mu_J, \sigma_J) \quad (17g)$$

$$u(t) = u, \quad \tilde{\xi}(t) = \tilde{\xi} \quad (17h)$$

In this equation, $\mathcal{N}(\cdot)$ describes a normal distribution, μ_k, μ_J , and $\mu_{\xi_{2,t_0}}$ are the mean values of the solar array stiffness, inertia ratio, and the initial condition for ξ_2 , respectively. Equation (17e) is included to ensure the feasibility of any boundary solution according to the discussion in Sec. (2.2). In this equation, k_s is the constraint shift index. Equation (17d) describes uncertain system dynamics, which are of Type I equality constraints [1]. Therefore, they must be satisfied “almost surely” or “a.s.” (with the probability of one). That is why these equations are described in terms of the infinite-dimensional uncertain quantities such as \tilde{k} and \tilde{J} . Finally, note that Eq. (17f) is only satisfied if an OLMC structure is used and relaxed for OLSC (not shown here).

4.2 Worst-Case Robust (WCR-UCCD)

When the probabilistic information of uncertain quantities is not available, a worst-case robust formulation provides a conservative solution that is optimal for the worst-case realization of uncertainties within the uncertainty set. Therefore, a major assumption in implementing such a formulation is the knowledge of the geometry and size of the uncertainty set. We assume that uncertainties belong to the following uncertainty set:

$$\mathcal{R}_q := \{ \tilde{q} \mid \|\delta_q\| \leq 1 \} \quad \text{where } \delta_q = \tilde{q} - \hat{q} \quad (18)$$

where \mathcal{R}_q is the uncertainty set, and \tilde{q} is a column vector composed of the concatenation of all uncertain quantities, and \hat{q} are nominal quantities that are used to construct the uncertainty sets. For the simple SASA problem, $\tilde{q}^T := [\tilde{k}, \tilde{J}, \tilde{\xi}_{2,t_0}]$. In addition, since all of the assumed uncertainties have a normal distribution, a natural choice in constructing the uncertainty set is to assume that $\hat{q} = \mu_q$ and $\delta_q = \pm k_s \sigma_q$, where $k_s = 3$. The epigraph representation of the worst-case robust formulation for the outer loop of the simple SASA problem is then described as [1]:

$$\underset{u, \mu_k, v}{\text{minimize:}} \quad -v \quad (19a)$$

$$\text{subject to: } v - \Phi(t, u, \tilde{\xi}, \tilde{k}(\mu_k), \tilde{J}, \tilde{\xi}_{2,t_0}) \leq 0 \quad (19b)$$

$$\text{Eqs. (16b)-(16c)} \quad (19c)$$

$$k_s \sigma_k - \mu_k \leq 0 \quad (19d)$$

$$\text{where: } u(t) = u \quad (19e)$$

where $u(t)$ and μ_k are inputs to the inner-loop optimization problem. In the outer-loop WCR-UCCD problem formulation, the number of optimization variables are $n_u = 1$ and $n_p = 2$. The

inner-loop WCR problem is formulated as:

$$\text{minimize: } o_{in} = \xi_1(t_f) \quad (20a)$$

$$\text{subject to: Eq. (16d)} \quad (20b)$$

$$\xi(t_0) = \begin{bmatrix} 0 \\ \xi_{2,t_0} \end{bmatrix} \quad (20c)$$

$$-0.3 \leq \xi_2(t_f) \leq 0.3 \quad (20d)$$

$$\text{where: } k(\mu_k) \in \mathcal{R}_k, \quad J(\mu_j) \in \mathcal{R}_j \quad (20e)$$

$$\xi_{2,t_0}(\mu_{\xi_{2,t_0}}) \in \mathcal{R}_{\xi_{2,t_0}}, \quad \xi(t) = \xi \quad (20f)$$

In this equation, the number of optimization variables include $n_s = 2$, and $n_p = 3$. Note that it is more natural for the WCR-UCCD formulation to be implemented through an OLS structure. That is why the above formulation requires the relaxation of the original terminal boundary condition, as shown in Eq. (20d). In this context, negative values are still physically meaningful because the state variables correspond to relative displacement and velocity.

A major limitation of this formulation is its inability to bring the spacecraft to rest. While this requirement is of utmost importance for the simple SASA problem, it may not be immediately realized through the OLS structure of WCR-UCCD. However, augmenting the inner-loop objective with a penalty term allows us to investigate the compromise between the competing objectives of the worst-case displacement and relative velocity at t_f . The inner-loop multiobjective optimization of the WCR-UCCD (equivalent to Eq. (20) but with different interpretations) can be formulated as:

$$\text{minimize: } o_{in} = \xi_1(t_f) + w_p w_n \xi_2^2(t_f) \quad (21a)$$

$$\text{subject to: Eqs. (20b)–(20c) and (20e)–(20f)} \quad (21b)$$

where $w_n = 10$ is a normalization factor that is included to ensure that both objective terms are of the same order of magnitude and w_p is the penalty weight. Varying the weight on the penalty term can then provide insights into the trade-offs between these competing objectives. Note that the associated outer-loop optimization problem generally remains the same. The only modification needed is in Eq. (19b) to make sure v is constrained only through $\xi_1(t_f)$ rather than the whole multiobjective value. This is to ensure that the epigraph variable is still representative of the associated $\xi_1(t_f)$. It is important to emphasize that this formulation offers a compromise between the desired objective of the poorest performance and the required terminal condition. As a result, this formulation may no longer correspond to the true worst-case robust solution. Even when $w_p = 0$, the solution will differ from the solution of Eq. (20) because in the former formulation, the terminal boundary conditions are completely relaxed, while they are only partially relaxed in Eq. (20).

TABLE 2: Settings for UCCD implementations of the simple SASA problem.

| Category | Option | Value |
|----------|-------------------|------------------|
| General | defects | trapezoidal (TR) |
| | mesh | equidistant |
| | quadrature | composite TR |
| | outer-loop solver | fmincon |
| | solver tolerance | 10^{-6} |
| SE-UCCD | inner-loop solver | quadprog |
| | derivatives | symbolic |
| | n_t | 100 |
| | N_{mcs} | 10,000 |
| | Q | 10^3 |
| WCR-UCCD | r_i | 8 |
| | M | 9^3 |
| | inner-loop solver | fmincon |
| | derivatives | forward |
| | n_t | 100 |

5 RESULTS AND DISCUSSION

The simple SASA UCCD was implemented and solved for the SE-UCCD and WCR-UCCD formulations introduced in Secs. 4.1 and 4.2. Since a nested UCCD coordination strategy results in linear dynamics and, thus, improved computational cost, this coordination strategy has been consistently used to obtain the solution of OLMC-SE-UCCD problem formulations. All of the inner-loop, optimal control subproblems from the nested coordination strategy are solved through a direct transcription (DT) approach. For all of the DT implementations, the MATLAB-based *DTQP* toolbox was used to efficiently construct and solve the UCCD problem [75, 76]. *DTQP* toolbox offers a simple problem definition approach, options for differentiation, and the possibility to take advantage of the linearized dynamics to obtain an efficient solution. A direct single shooting (DSS) approach is used for the outer-loop optimization problem of WCR-UCCD, while the inner-loop is solved using *DTQP*. The SE-UCCD formulation was solved using MCS and gPC. All of the results and implementations are made available in Ref. [74].

The computer architecture used for these case studies is a desktop workstation with an AMD 3900X 12-core processor at 3.79 GHz, 32 GB of RAM, 64-bit windows 10 Enterprise LTSC version 1809, and MATLAB R2021b. The settings associated with the *DTQP* toolbox, optimization solvers, and specific method-dependent parameters for MCS and gPC are reported in Table 2. The results from these UCCD implementations for the simple SASA problem are presented in Table 3. Note that in this table, \bar{o} is the objective function associated with the corresponding formulation introduced in Eqs. (17a) and (19a), respectively.

TABLE 3: Summary of SE-UCCD and WCR-UCCD solutions.

| Form. ¹ | Str. ² | $\bar{\sigma}$ | μ_k | $t(s)$ | t_{switch} |
|--------------------|-------------------|----------------|---------|--------|--------------|
| CCD | OLSC | -0.301 | 3.441 | 3 | 0.727 |
| Stc-MCS | OLMC | -0.308 | 3.311 | 5717 | 0.737 |
| Stc-gPC | OLMC | -0.306 | 3.185 | 562 | 0.742 |
| $ \Delta $ | - | 0.65% | 3.81% | 90% | 0.68% |
| WcR | OLSC | 0.204 | 0.705 | 2592 | 0.838 |

¹ Formulation, ² Control structure under uncertainties

5.1 SE-UCCD Solution

According to Table 2, OLMC stochastic results entail the solution of an MCS and gPC UP methods. Specifically, using $N_{mcs} = 10,000$ number of samples for the MCS-based stochastic UCCD problem, the solution (which is described in Table 3) converged to the objective value of $\bar{\sigma} = -0.308$, with $\bar{k} = 3.311$. This solution offers a relatively small error of $O(0.01)$ for the objective function and thus, is used to benchmark results from gPC for OLMC implementations. The solution from gPC differs from that of MCS by 0.65%, and 3.81% for $\bar{\sigma}$ and \bar{k} , respectively. However, the computational time is reduced by 90.17%, which is a considerable improvement over the MCS method. Since the optimal behavior of the system is bang-bang control with a single switch, the average control switching time t_{switch} was also estimated for all of these implementations. From Table 3, the average control switching time for gPC differs from that of MCS by 0.68%. It is expected that the results from gPC can be further improved by increasing the number of nodes, particularly in the dimension of \bar{k} .

Compared to the deterministic solution, the results indicate an increase in the expected value of the objective function. This increase is justified by the fact that the stochastic expected value model is a risk-neutral formulation in which no measure is taken to move away from uncertainties. Therefore, depending on their distribution and the corresponding model, some uncertainty realizations may act in favor of the objective function. Therefore, the objective function may witness an increase or a decrease compared to the deterministic solution. Note, however, that this is generally not the case for stochastic chance-constrained formulations because their risk-averse nature pushes the solutions towards the interior of the feasible space to maintain the desired probability of success.

Optimal state and control trajectories, along with their associated $\pm\sigma$ distribution bands for MCS and gPC-based implementations, are shown in Fig. 4 and compared with the deterministic solution. The results indicate that gPC is very well capable of estimating statistics of problem elements. Specifically, according to Figs. 4a and 4c, the expected value response for optimal state trajectories from both MCS and gPC-based methods are slightly higher than the their associated deterministic counterparts. This observation is also aligned with the increase in the objective

function value. In addition, the OLMC structure allows the optimizer to change the controller's switching time in response to the realization of uncertainties. This results in a range of optimal control responses whose distribution band with $\pm\sigma_{\bar{u}}$ is shown in Figs. 4b and 4d for MCS and gPC, respectively. Note that while the control is expected to remain bang-bang in nature, the resulting trajectories (with intermediate control quantities) seem to be a by-product of the optimization algorithm and/or UP methods. In addition, note that the average control switching time signifies a 1.37% difference between MCS and deterministic solutions, which points to the utility of OLMC structure as it allows the simultaneous exploration of plant and control spaces to find the solutions in the presence of uncertainties.

5.2 WCR-UCCD Solution

Acquiring the optimal control response for OLSC-WCR-UCCD problem based on Eq. (19) requires a DSS approach. Using 100 collocation points, we implemented and solved the problem with the inner-loop formulation of Eq. (20) and tabulated the results in Table 3. The solution converged in 2592 s to the outer-loop objective value of 0.204 with $\mu_k = 0.705$. This solution differs from the deterministic solution by 32.23% for the objective function and 79.5% for solar array stiffness. In addition, the average switching time has increased by 15.27% compared to the deterministic case. When compared with the risk-neutral SE-UCCD, the results highlight three important observations about WCR-UCCD formulation: (i) the conservativeness of the solution, (ii) the risk-averse nature of the formulation, and (iii) inability to bring the vehicle to stop at t_f . In addition, note that while the computational time of the WCR-UCCD formulation is significantly less than the SE-UCCD with the MCS approach, it is still much higher than gPC.

The optimal control and state trajectories associated with OLSC-WCR-UCCD solution are shown in Fig. 5. According to Fig. 5a, $\xi_{1,WCR}(t)$ is much lower than the deterministic case. Note, however, that $\xi_2(t_f)$ in the WCR solution does not approach 0 because, according to Eq. (20d), the terminal boundary condition was partially relaxed. The optimal control trajectory associated with WCR is shown in Fig. 5b and compared with the deterministic case. Note that while the optimal control trajectory is expected to remain bang-bang in nature, the resulting WCR trajectory indicates a slight deviation from this expected behavior. This issue seems to be directly related to the limitations of the OLSC optimization algorithm. From this figure, it is evident that the average switching time of the worst-case robust optimal control trajectory is delayed compared to the deterministic case.

Implementing WCR-UCCD using the multiobjective formulation of Eq. (21) with various penalty weights w_p allows us to investigate the trade-offs between the worst-case (minimum) relative displacement and the requirement of bringing the vehicle to rest at t_f . In the limit, these implementations correspond to the worst-case robust solution when $w_p = 0$ (but with com-

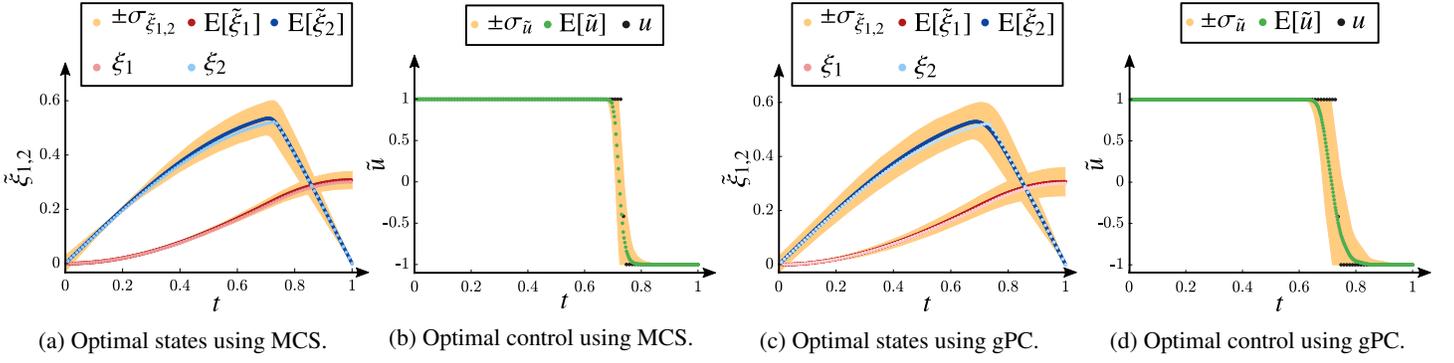


FIGURE 4: Open-Loop multiple-control (OLMC), stochastic in expectation UCCD (SE-UCCD) solution using MCS and gPC in comparison to the deterministic CCD solution.

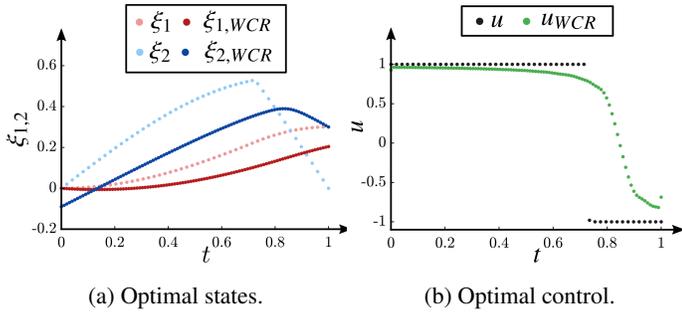


FIGURE 5: Worst-case robust solution of the OLSA simple SASA UCCD problem compared to the deterministic CCD solution.

pletely relaxed terminal conditions—thus different from the results shown in Fig. 5), and the case of bringing the vehicle to rest when $w_p = 1$. The results from this implementation are shown in Fig. 6. It is evident that the solution obtained from these implementations does not always correspond to the true worst-case realization of uncertainties; rather, it is a compromise between the two competing objectives.

When $w_p = 0$, the terminal condition is completely relaxed and the optimal control trajectory remains relatively constant at $u(t) = 1$. Interestingly, this behavior remains the same for some other weighting factors, including the case of $w_p = 1$. For the remaining weighting factors, no general trend is witnessed. In all of these cases, the optimal control trajectories are no longer bang-bang in nature. These results are more difficult to interpret and are inconsistent with our expectation that increasing the penalty weight should result in poorer performance and lower final velocity. From this implementation, it is reasonable to state that the multiobjective OLSA formulation is less numerically robust compared to the OLSA formulation introduced in Eq. (20), indicating future work is needed to better understand these trends.

Next, the OLMC-WCR-UCCD problem is investigated using the concept of polytopic uncertainties. A polytope is a bounded, closed, and convex polyhedron described by a finite number of affine inequalities. For a linear program, the feasible region of the optimization problem is a convex polytope, (i.e., the convex hull of the vertices of the polytope), and the optimal solution is achieved at a vertex. Therefore, we only need to check the vertices of the polytope. Since the OLMC-WCR-UCCD problem has polytopic uncertainties and is linear when using a nested coordination strategy, then the aforementioned results from linear programming may be applied. In other words, the number of required evaluations can be reduced to function evaluations at the vertices of the polytope. This property was also confirmed by sweeping the uncertain space and ascertaining that the worst-case combination of uncertainties is achieved at a vertex.

Next, we investigate the solution of OLMC-WCR-UCCD by substituting the inner-loop robust optimization problem with function evaluations at the vertices of the polytope. The results from this implementation are shown in Fig. 7 and compared with the deterministic CCD solution. Specifically, Fig. 7a illustrates the polytope formed by uncertainty sets, with a total of 2^3 vertices. In these figures, all gray trajectories are associated with vertices that do not represent the worst-case realization of uncertainties. Figure 7b shows ξ_1 trajectories for all of the vertices of the polytope. Note that while there is a total of 8 vertices in the polytope, some trajectories completely overlap with others (due to problem symmetry), resulting in a total of 4 distinct trajectories for states. Each pair of overlapping trajectories have J and ξ_{2,t_0} in common, but differ in k dimension. Since μ_k is an optimization variable, the optimizer selects this value such that the total stiffness in the optimization problem amounts to the same value—resulting in identical trajectories.

The worst-case performance of the system corresponds to the combination of $(\mu_k \pm k_s \sigma_k, -k_s \sigma_{\xi_{2,t_0}}, k_s \sigma_J)$ with $\xi_1(t_f) = 0.157$. According to Fig. 7c, ξ_2 trajectories associated with the worst-case follow a lower (slower) path compared to the deter-

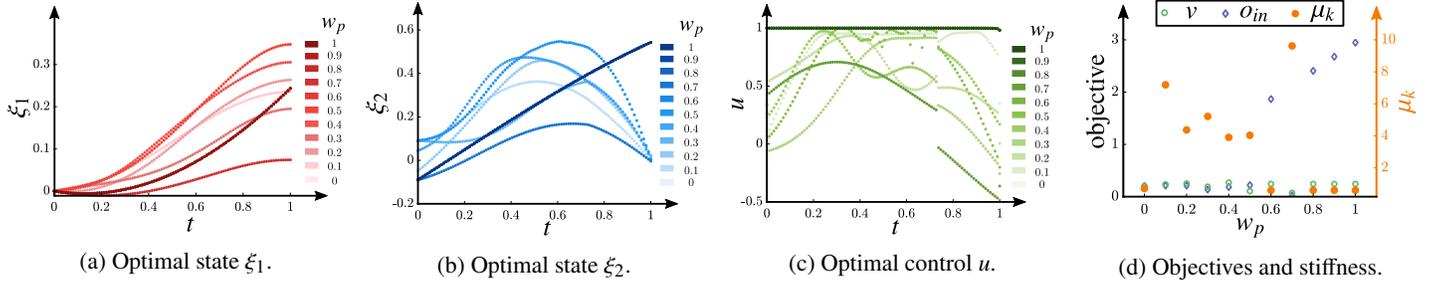


FIGURE 6: Open-loop single-control (OLSC) implementation of WCR-UCCD using a multiobjective inner-loop optimization problem with different penalty weights for the simple SASA problem (Eqs. (19) and (21)).

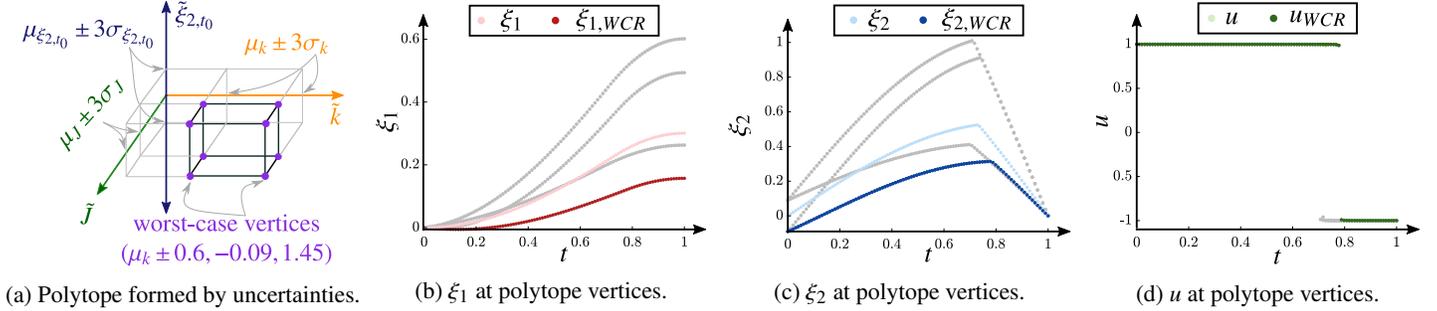


FIGURE 7: Open-loop multiple-control (OLMC) implementation of the WCR-UCCD for the simple SASA problem using polytopic uncertainties.

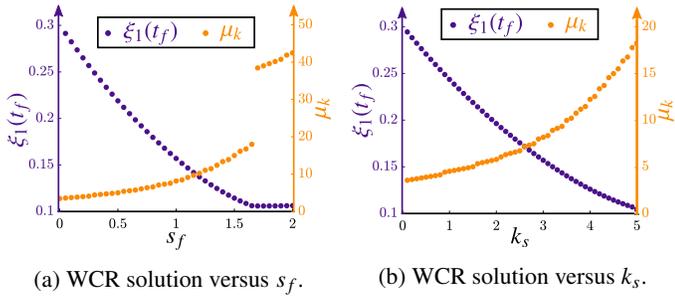


FIGURE 8: OLMC-WCR-UCCD objective and solar array stiffness as a function of (a) s_f which scales standard deviation of uncertainties, and (b) k_s which characterizes the size of the uncertainty set.

ministic case. Optimal control trajectories associated with the vertices of the polytope are shown in Fig. 7d. Although difficult to see, these trajectories have different switching times. These implementations indicate that the exploration of the whole design spaces (both plant and control) through OLMC structures reduces some of the issues with numerical stability, as well as meeting problem requirements in the OLS structure.

It is clear from Fig. 7 that uncertainties in this case study are relatively large because, under the worst realization of uncertainties, the performance of the system is significantly affected

by 47.84%. Invaluable insights from this observation can then be offered to the design team or manufacturer. For example, by investing in higher-precision equipment and measurement devices or improving the precision of the manufacturing processes, the manufacturer may, to some extent, reduce these uncertainties and, therefore, improve the worst-case performance of the system. To illustrate this, we use a size factor, s_f , to scale (up/down) the standard deviation of uncertainties from their current values, such that $\hat{\sigma}_q = s_f \sigma_q$, where $\hat{\sigma}_q$ is the new standard deviation. Therefore, $s_f = 1$ correspond to the current standard deviations, $s_f < 1$ corresponds to smaller standard deviations, and $s_f > 1$ corresponds to larger standard deviations for uncertainties.

The objective function, $\xi_1(t_f)$ and solar array stiffness, μ_k for different s_f values are shown in Fig. 8a. From this figure, we can see that as uncertainties are increased, the performance of the system drops down until it reaches a small value of 0.1. In addition, note that there's a sudden jump in the optimal value of μ_k when $s_f = 1.6$. In addition to the standard deviation of uncertainties, the deterministic uncertainty set is also characterized through k_s . As opposed to s_f , which represents the inherent magnitude of uncertainties, k_s reflects designers' understanding of the size of the uncertainty set. In other words, k_s is selected such that the resulting uncertainty set is a close approximation of uncertainty realizations. If k_s is too small, the resulting uncertainty set will not be a good representation, and thus, the WCR-UCCD

solution will not be able to protect the system against risks associated with uncertainties. On the other hand, when k_s is too large, the solution will be over-conservative, and thus, the system may be subject to low performance due to over-design. To investigate this issue, Fig. 8b shows the objective function and solar array stiffness values as functions of k_s when $s_f = 1$ (i.e., current magnitude of uncertainties).

It is clear from these investigations that uncertainties should be treated as a critical element in dynamic system design from the early stages and not as an afterthought. Exploration of the whole design space (both plant and control) at the early stages of the design process, through OLMC structures and appropriate UCCD formulations, enables designers to understand the general trends and interactions between various system elements and have reasonable expectations of the system performance under uncertainties.

6 CONCLUSION

Implementation of uncertain control co-design (UCCD) formulations for real-world dynamic systems requires an in-depth understanding of the interpretation of each formulation, uncertainty organizational schemes, and solution strategies. Therefore, in this article, we introduced the open-loop single control (OLSC) and open-loop multiple control (OLMC) structures under uncertainties to ensure meeting problem-specific requirements. We also described two uncertainty propagation methods for probabilistic uncertainties and solved a simple UCCD case study with the stochastic in expectation UCCD (SE-UCCD) and worst-case robust UCCD (WCR-UCCD) formulations. By doing so, this study seeks to provide a foundation for future advances in UCCD problems.

The SE-UCCD problem was solved using MCS and gPC in a nested coordination strategy. The results from this implementation indicate that gPC offers promising improvements in the computational time of UCCD problems. In addition, the risk-neutral nature of this formulation results in an objective function that is slightly higher than that of the deterministic solution. For WCR-UCCD, uncertainties belonged to a crisp uncertainty set and were characterized based on $\pm 3\sigma$ from the mean values. While OLSC structure is more natural for WCR-UCCD, we also investigated OLMC solution using polytopic uncertainties. We observed that OLSC solutions are not as numerically robust (as currently implemented) as those obtained using an OLMC structure. The results obtained in this study using various formulations, organizational structure of uncertainties, and solution strategies indicate that uncertainty considerations impact system and design judgment and can be a critical element in the early stages of the design process.

A natural next step is to extend this work to investigate problems with probabilistic path constraints, with an emphasis on stochastic chance-constraints UCCD formulations. The in-

clusion of time-dependent disturbances in the dynamic system model, along with its uncertainty propagation through an efficient and effective method, is a crucial step in implementing UCCD formulations for real-world applications. The implementation of Galerkin type of gPC, which is promising for problems with a small number of state variables, might provide an improvement over the collocation type of gPC for small-scale problems. Comparison with the most-probable-point (MPP)-based UP methods, which have been already studied in reliability-based CCD, may offer additional insights into possible improvements in computational time, especially because the number of dynamic system equations can be reduced significantly. For WCR-UCCD, it is essential to consider various geometries (such as ellipsoidal, hexagonal, etc.) for the uncertainty set. Since empirical information about uncertainties is generally limited in early-stage design, it is necessary to investigate non-probabilistic propagation methods such as interval analysis and methods from fuzzy programming. Finally, the implementations of such methods and formulation for larger problems must be investigated to address the question of scalability.

REFERENCES

- [1] Azad, S., and Herber, D. R., 2022. "Control co-design under uncertainties: formulations". In International Design Engineering Technical Conferences, no. DETC2022-89507.
- [2] Ruszczyński, A., and Shapiro, A., 2003. "Stochastic programming models". *Handbooks Oper. Res. Management Sci.*, **10**, pp. 1–64. doi: [10.1016/S0927-0507\(03\)10001-1](https://doi.org/10.1016/S0927-0507(03)10001-1)
- [3] Powell, W. B., 2019. "A unified framework for stochastic optimization". *Eur. J. Oper. Res.*, **275**(3), pp. 795–821. doi: [10.1016/j.ejor.2018.07.014](https://doi.org/10.1016/j.ejor.2018.07.014)
- [4] Beyer, H.-G., and Sendhoff, B., 2007. "Robust optimization—a comprehensive survey". *Comput. Methods Appl. Mech. Eng.*, **196**(33–34), pp. 3190–3218. doi: [10.1016/j.cma.2007.03.003](https://doi.org/10.1016/j.cma.2007.03.003)
- [5] Gorissen, B. L., Yanikoğlu, İ., and den Hertog, D., 2015. "A practical guide to robust optimization". *Omega*, **53**, pp. 124–137. doi: [10.1016/j.omega.2014.12.006](https://doi.org/10.1016/j.omega.2014.12.006)
- [6] Bertsimas, D., Brown, D. B., and Caramanis, C., 2011. "Theory and applications of robust optimization". *SIAM Rev.*, **53**(3), pp. 464–501. doi: [10.1137/080734510](https://doi.org/10.1137/080734510)
- [7] Zadeh, L. A., 1996. "Fuzzy sets". In *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*. World Scientific, pp. 394–432. doi: [10.1142/9789814261302_0021](https://doi.org/10.1142/9789814261302_0021)
- [8] Zadeh, L. A., 1978. "Fuzzy sets as a basis for a theory of possibility". *Fuzzy Sets Syst.*, **1**(1), pp. 3–28. doi: [10.1016/0165-0114\(78\)90029-5](https://doi.org/10.1016/0165-0114(78)90029-5)
- [9] Liu, B., and Liu, B., 2009. *Theory and Practice of Uncertain Programming*. Springer. doi: [10.1007/978-3-540-89484-1](https://doi.org/10.1007/978-3-540-89484-1)
- [10] Herber, D. R., and Allison, J. T., 2019. "Nested and simultaneous solution strategies for general combined plant and control design problems". *J. Mech. Design*, **141**(1). doi: [10.1115/1.4040705](https://doi.org/10.1115/1.4040705)
- [11] Allison, J. T., and Herber, D. R., 2014. "Multidisciplinary design optimization of dynamic engineering systems". *AIAA J.*, **52**(4), pp. 691–710. doi: [10.2514/1.J052182](https://doi.org/10.2514/1.J052182)
- [12] Sundararajan, A. K., and Herber, D. R., 2021. "Towards a fair comparison between the nested and simultaneous control co-design methods using an active suspension case study". In American Control Conference, pp. 358–365. doi: [10.23919/ACC50511.2021.9482687](https://doi.org/10.23919/ACC50511.2021.9482687)

- [13] Diehl, M., Gerhard, J., Marquardt, W., and Mönnigmann, M., 2008. “Numerical solution approaches for robust nonlinear optimal control problems”. *Comput. Chem. Eng.*, **32**(6), pp. 1279–1292. doi: [10.1016/j.compchemeng.2007.06.002](https://doi.org/10.1016/j.compchemeng.2007.06.002)
- [14] Diehl, M., Bock, H. G., and Kostina, E., 2006. “An approximation technique for robust nonlinear optimization”. *Math. Program.*, **107**(1), pp. 213–230. doi: [10.1007/s10107-005-0685-1](https://doi.org/10.1007/s10107-005-0685-1)
- [15] Ma, D. L., and Braatz, R. D., 2001. “Worst-case analysis of finite-time control policies”. *IEEE Trans. Control Syst. Technol.*, **9**(5). doi: [10.1109/87.944471](https://doi.org/10.1109/87.944471)
- [16] Nash, A. L., Pangborn, H. C., and Jain, N., 2021. “Robust control co-design with receding-horizon MPC”. In American Control Conference, pp. 373–379. doi: [10.23919/ACC50511.2021.9483216](https://doi.org/10.23919/ACC50511.2021.9483216)
- [17] Nakka, Y. K., and Chung, S.-J., 2021. Trajectory optimization of chance-constrained nonlinear stochastic systems for motion planning and control. arXiv:2106.02801
- [18] Berning, A. W., Taheri, E., Girard, A., and Kolmanovsky, I., 2018. “Rapid uncertainty propagation and chance-constrained trajectory optimization for small unmanned aerial vehicles”. In American Control Conference, pp. 3183–3188. doi: [10.23919/ACC.2018.8431765](https://doi.org/10.23919/ACC.2018.8431765)
- [19] Herber, D. R., and Allison, J. T., 2017. “Unified scaling of dynamic optimization design formulations”. In International Design Engineering Technical Conferences, no. DETC2017-67676. doi: [10.1115/DETC2017-67676](https://doi.org/10.1115/DETC2017-67676)
- [20] Chilan, C. M., Herber, D. R., Nakka, Y. K., Chung, S.-J., Allison, J. T., Aldrich, J. B., and Alvarez-Salazar, O. S., 2017. “Co-design of strain-actuated solar arrays for spacecraft precision pointing and jitter reduction”. *AIAA J.*, **55**(9). doi: [10.2514/1.J055748](https://doi.org/10.2514/1.J055748)
- [21] Herber, D. R., and Sundarajan, A. K., 2020. “On the uses of linear-quadratic methods in solving nonlinear dynamic optimization problems with direct transcription”. In ASME International Mechanical Engineering Congress and Exposition, p. V07AT07A003. doi: [10.1115/IMECE2020-23885](https://doi.org/10.1115/IMECE2020-23885)
- [22] Fathy, H. K., Papalambros, P. Y., Ulsoy, A. G., and Hrovat, D., 2003. “Nested plant/controller optimization with application to combined passive/active automotive suspensions”. In American Control Conference, pp. 3375–3380. doi: [10.1109/ACC.2003.1244053](https://doi.org/10.1109/ACC.2003.1244053)
- [23] Allison, J. T., Guo, T., and Han, Z., 2014. “Co-design of an active suspension using simultaneous dynamic optimization”. *J. Mech. Design*, **136**(8). doi: [10.1115/1.4027335](https://doi.org/10.1115/1.4027335)
- [24] Betts, J. T., 2010. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM. doi: [10.1137/1.9780898718577](https://doi.org/10.1137/1.9780898718577)
- [25] Biegler, L. T., 2010. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM. doi: [10.1137/1.9780898719383](https://doi.org/10.1137/1.9780898719383)
- [26] Rao, A. V., 2009. “A survey of numerical methods for optimal control”. *Adv. Astronaut. Sci.*, **135**(1), pp. 497–528.
- [27] Azad, S., and Alexander-Ramos, M. J., 2020. “A single-loop reliability-based MDSO formulation for combined design and control optimization of stochastic dynamic systems”. *J. Mech. Design*, **143**(2), p. 021703. doi: [10.1115/1.4047870](https://doi.org/10.1115/1.4047870)
- [28] Azad, S., and Alexander-Ramos, M. J., 2021. “Robust combined design and control optimization of hybrid-electric vehicles using MDSO”. *IEEE Trans. Veh. Technol.*, **70**(5), pp. 4139–4152. doi: [10.1109/TVT.2021.3071863](https://doi.org/10.1109/TVT.2021.3071863)
- [29] Fisher, J., and Bhattacharya, R., 2011. “Optimal trajectory generation with probabilistic system uncertainty using polynomial chaos”. *J. Dyn. Syst. Meas. Control*, **133**(1). doi: [10.1115/1.4002705](https://doi.org/10.1115/1.4002705)
- [30] Boutselis, G. I., De La Torre, G., and Theodorou, E. A., 2016. “Stochastic optimal control using polynomial chaos variational integrators”. In American Control Conference, pp. 6586–6591. doi: [10.1109/ACC.2016.7526707](https://doi.org/10.1109/ACC.2016.7526707)
- [31] Cottrill, G. C., 2012. “Hybrid solution of stochastic optimal control problems using Gauss pseudospectral method and generalized polynomial chaos algorithms”. PhD thesis, Air Force Institute of Technology.
- [32] Nagy, Z. K., and Braatz, R. D., 2004. “Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis”. *J. Process Control*, **14**(4). doi: [10.1016/j.jprocont.2003.07.004](https://doi.org/10.1016/j.jprocont.2003.07.004)
- [33] Meyers, J. J., Leonard, A. M., Rogers, J. D., and Gerlach, A. R., 2019. “Koopman operator approach to optimal control selection under uncertainty”. In American Control Conference. doi: [10.23919/ACC.2019.8814461](https://doi.org/10.23919/ACC.2019.8814461)
- [34] González Arribas, D., Soler Arnedo, M. F., and Sanjurjo Rivo, M., 2016. In Proceedings of SESAR Innovation Days 2016.
- [35] Behtash, M., and Alexander-Ramos, M. J., 2021. “A reliability-based formulation for simulation-based control co-design using generalized polynomial chaos expansion”. *J. Mech. Design*, **144**(5), p. 051705. doi: [10.1115/1.4052906](https://doi.org/10.1115/1.4052906)
- [36] Mattson, C., and Messac, A., 2003. “Handling equality constraints in robust design optimization”. In AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. doi: [10.2514/6.2003-1780](https://doi.org/10.2514/6.2003-1780)
- [37] Rangavajhala, S., Mullur, A., and Messac, A., 2007. “The challenge of equality constraints in robust design optimization: examination and new approach”. *Struct. Multidiscipl. Optim.*, **34**(5), pp. 381–401. doi: [10.1007/s00158-007-0104-8](https://doi.org/10.1007/s00158-007-0104-8)
- [38] Azad, S., 2020. “Combined design and control optimization of stochastic dynamic systems”. PhD thesis, University of Cincinnati.
- [39] Kelly, M., 2017. “An introduction to trajectory optimization: How to do your own direct collocation”. *SIAM Rev.*, **59**(4), pp. 849–904. doi: [10.1137/16M1062569](https://doi.org/10.1137/16M1062569)
- [40] Kitapbayev, Y., Moriarty, J., and Mancarella, P., 2015. “Stochastic control and real options valuation of thermal storage-enabled demand response from flexible district energy systems”. *Appl. Energy*, **137**. doi: [10.1016/j.apenergy.2014.07.019](https://doi.org/10.1016/j.apenergy.2014.07.019)
- [41] Chai, R., Savvaris, A., Tsourdos, A., Chai, S., Xia, Y., and Wang, S., 2019. “Solving trajectory optimization problems in the presence of probabilistic constraints”. *IEEE Trans. Cybern.*, **50**(10). doi: [10.1109/TCYB.2019.2895305](https://doi.org/10.1109/TCYB.2019.2895305)
- [42] Cottrill, G., and Harmon, F., 2011. “Hybrid Gauss pseudospectral and generalized polynomial chaos algorithm to solve stochastic trajectory optimization problems”. In AIAA Guidance, Navigation, and Control Conference. doi: [10.2514/6.2011-6572](https://doi.org/10.2514/6.2011-6572)
- [43] Matsuno, Y., and Tsuchiya, T., 2013. “4D trajectory optimization in the presence of uncertainty”. In Aviation Technology, Integration, and Operations Conference. doi: [10.2514/6.2013-4323](https://doi.org/10.2514/6.2013-4323)
- [44] Cui, T., Allison, J. T., and Wang, P., 2020. “A comparative study of formulations and algorithms for reliability-based co-design problems”. *J. Mech. Design*, **142**(3). doi: [10.1115/1.4045299](https://doi.org/10.1115/1.4045299)
- [45] Azad, S., and Alexander-Ramos, M. J., 2020. “Robust MDSO for co-design of stochastic dynamic systems”. *J. Mech. Design*, **142**(1). doi: [10.1115/1.4044430](https://doi.org/10.1115/1.4044430)
- [46] Parkinson, A., Sorensen, C., and Pourhassan, N., 1993. “A general approach for robust optimal design”. *J. Mech. Design*, **115**(1). doi: [10.1115/1.2919328](https://doi.org/10.1115/1.2919328)
- [47] Liu, J., Meng, X., Xu, C., Zhang, D., and Jiang, C., 2018. “Forward and inverse structural uncertainty propagations under stochastic variables with arbitrary probability distributions”. *Comput. Methods Appl. Mech. Eng.*, **342**. doi: [10.1016/j.cma.2018.07.035](https://doi.org/10.1016/j.cma.2018.07.035)
- [48] Litvinenko, A., and Matthies, H. G., 2013. Inverse problems and uncertainty quantification. arXiv:1312.5048v2
- [49] Fishman, G., 2013. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer. doi: [10.1007/978-1-4757-2553-7](https://doi.org/10.1007/978-1-4757-2553-7)
- [50] Liu, W. K., Belytschko, T., and Mani, A., 1986. “Probabilistic finite elements for nonlinear structural dynamics”. *Comput. Methods Appl. Mech. Eng.*, **56**(1). doi: [10.1016/0045-7825\(86\)90136-2](https://doi.org/10.1016/0045-7825(86)90136-2)

- [51] Stefanou, G., 2009. “The stochastic finite element method: past, present and future”. *Comput. Methods Appl. Mech. Eng.*, **198**(9-12). doi: [10.1016/j.cma.2008.11.007](https://doi.org/10.1016/j.cma.2008.11.007)
- [52] Holmes, M. H., 2012. *Introduction to Perturbation Methods*. Springer. doi: [10.1007/978-1-4614-5477-9](https://doi.org/10.1007/978-1-4614-5477-9)
- [53] Ghanem, R. G., and Spanos, P. D., 2003. *Stochastic Finite Elements: A Spectral Approach*. Springer. doi: [10.1007/978-1-4612-3094-6](https://doi.org/10.1007/978-1-4612-3094-6)
- [54] Xiu, D., 2009. “Fast numerical methods for stochastic computations: a review”. *Commun. Comput. Phys.*, **5**(2-4).
- [55] Gunzburger, M. D., Lee, H.-C., and Lee, J., 2011. “Error estimates of stochastic optimal Neumann boundary control problems”. *SIAM J. Numer. Anal.*, **49**(4), pp. 1532–1552. doi: [10.1137/100801731](https://doi.org/10.1137/100801731)
- [56] Du, X., and Chen, W., 2004. “Sequential optimization and reliability assessment method for efficient probabilistic design”. *J. Mech. Design*, **126**(2). doi: [10.1115/1.1649968](https://doi.org/10.1115/1.1649968)
- [57] Youn, B. D., Choi, K. K., and Du, L., 2005. “Enriched performance measure approach for reliability-based design optimization”. *AIAA J.*, **43**(4). doi: [10.2514/1.6648](https://doi.org/10.2514/1.6648)
- [58] Zhang, J., and Du, X., 2010. “A second-order reliability method with first-order efficiency”. *J. Mech. Design*, **132**(10). doi: [10.1115/1.4002459](https://doi.org/10.1115/1.4002459)
- [59] Lee, S. H., and Chen, W., 2009. “A comparative study of uncertainty propagation methods for black-box-type problems”. *Struct. Multidiscipl. Optim.*, **37**(3). doi: [10.1007/s00158-008-0234-7](https://doi.org/10.1007/s00158-008-0234-7)
- [60] Seo, H. S., and Kwak, B. M., 2002. “Efficient statistical tolerance analysis for general distributions using three-point information”. *Int. J. Prod. Res.*, **40**(4). doi: [10.1080/00207540110095709](https://doi.org/10.1080/00207540110095709)
- [61] Huyse, L., and Walters, R. W., 2001. Random field solutions including boundary condition uncertainty for the steady-state generalized Burgers equation. Tech. rep., National Aeronautics and Space Administration Hampton VA Langley Research Center.
- [62] McGraw, R., 1997. “Description of aerosol dynamics by the quadrature method of moments”. *Aerosol Sci. Tech.*, **27**(2). doi: [10.1080/02786829708965471](https://doi.org/10.1080/02786829708965471)
- [63] Kumar, M., Singla, P., Chakravorty, S., and Junkins, J. L., 2006. “A multi-resolution approach for steady-state uncertainty determination in nonlinear dynamical systems”. In Southeastern Symposium on System Theory. doi: [10.1109/SSST.2006.1619059](https://doi.org/10.1109/SSST.2006.1619059)
- [64] Simon, D., 2006. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons. doi: [10.1002/0470045345](https://doi.org/10.1002/0470045345)
- [65] Zio, E., and Pedroni, N., 2013. Literature review of methods for representing uncertainty. Tech. rep., Fondation pour une Culture de Sécurité Industrielle.
- [66] Shapiro, A., Dentcheva, D., and Ruszczyński, A., 2021. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM. doi: [10.1137/1.9781611976595](https://doi.org/10.1137/1.9781611976595)
- [67] Xiu, D., 2010. *Numerical Methods for Stochastic Computations*. Princeton University Press.
- [68] Loeve, M., 1978. *Probability Theory II*, 4th ed. Springer.
- [69] Rosenblatt, M., 1952. “Remarks on a multivariate transformation”. *Ann. Math. Stat.*, **23**(3). doi: [10.1214/aoms/1177729394](https://doi.org/10.1214/aoms/1177729394)
- [70] Wang, F., Yang, S., Xiong, F., Lin, Q., and Song, J., 2019. “Robust trajectory optimization using polynomial chaos and convex optimization”. *Aerosp. Sci. Technol.*, **92**, pp. 314–325. doi: [10.1016/j.ast.2019.06.011](https://doi.org/10.1016/j.ast.2019.06.011)
- [71] Wang, F., Xiong, F., Jiang, H., and Song, J., 2018. “An enhanced data-driven polynomial chaos method for uncertainty propagation”. *Eng. Optim.*, **50**(2). doi: [10.1080/0305215X.2017.1323890](https://doi.org/10.1080/0305215X.2017.1323890)
- [72] Künzner, F., 2020. “Efficient non-intrusive uncertainty quantification for large-scale simulation scenarios”. PhD thesis, Universität München.
- [73] Smolyak, S. A., 1963. “Quadrature and interpolation formulas for tensor products of certain classes of functions”. *Dokl. Akad. Nauk SSSR*, **148**(5).
- [74] <https://github.com/AzadSaeed/uncertain-control-co-design-SASA-example.git>.
- [75] Herber, D. R., 2017. “Advances in combined architecture, plant, and control design”. PhD thesis, University of Illinois at Urbana-Champaign.
- [76] The DTQP Project. <https://github.com/danielrherber/dt-qp-project>.

Nomenclature

Acronyms

| | |
|----------|--|
| a.s. | almost surely |
| AAO | all-at-once |
| CCD | control co-design |
| CTR | composite trapezoidal rule |
| DSS | direct single shooting |
| DT | direct transcription |
| DTQP | direct transcription quadratic programming |
| ED | equidistant (nodes) |
| FE-UCCD | fuzzy expected value UCCD |
| FORM | first-order reliability method |
| FOSM | first-order second moment |
| gPC | generalized polynomial chaos |
| MCS | Monte Carlo simulation |
| MPP | most-probable-point |
| ODE | ordinary differential equations |
| OLMC | open-loop multiple control |
| OLSC | open-loop single control |
| PCC-UCCD | possibilistic chance-constrained UCCD |
| PDF | probability distribution function |
| PR-UCCD | probabilistic robust UCCD |
| SASA | strain-actuated solar array |
| SCC-UCCD | stochastic chance-constrained UCCD |
| SE-UCCD | stochastic in expectation UCCD |
| SORM | second-order reliability method |
| TR | trapezoidal rule |
| UCCD | uncertain control co-design |
| UP | uncertainty propagation |
| WCR-UCCD | worst-case robust UCCD |

Notation

| | |
|------------------------|---|
| α_w | quadrature weights |
| $\bar{\bullet}(\cdot)$ | composite function of $\bullet(\cdot)$ |
| $\xi(t)$ | state trajectories |
| ξ_0 | initial conditions |
| ξ_f | terminal conditions |
| $f(\cdot)$ | state transition or derivative function |
| $g(\cdot)$ | inequality constraint vector |
| $h(\cdot)$ | equality constraint vector |
| p | time-independent optimization variable vector |
| p_c | control gain vector |
| p_p | plant optimization variables |
| $u(t)$ | open-loop control trajectory vector |

| | | | |
|--------------------------------|--|--------------------------------------|--|
| δ | Kronecker delta function | $\tilde{y}(\cdot)$ | an arbitrary second-order variable or process |
| $\dot{\bullet}$ | time-derivative of \bullet | $F_{\tilde{\mathbf{x}}}(\mathbf{x})$ | joint probability distribution |
| $\ell(\cdot)$ | Lagrange term | $H_k(\tilde{x}_i)$ | Hermite polynomial basis functions |
| Γ | finite domain of distribution function | J | inertia ratio between the solar array and the bus |
| γ | normalization factor | K | solar array stiffness |
| $\hat{\mathbf{q}}$ | nominal quantities used to construct uncertainty sets | k_s | constraint shift index |
| $\hat{\mathbb{P}}_{f,i}$ | estimate of the probability of failure for i th constraint | $m(\cdot)$ | Mayer term |
| $\hat{\mu}$ | unbiased estimate of the mean of $\tilde{\bullet}$ | M | dimension of the n_x -variate polynomials |
| $\hat{\sigma}^2$ | unbiased estimate of the variance of $\tilde{\bullet}$ | N_d | last element in the support of some discrete distributions |
| $\hat{y}_m(\cdot)$ | gPC expansion coefficients | n_d | number of elements in the vector of problem data |
| $\mathbb{E}[\cdot]$ | expected value operator | N_{mcs} | number of samples in MCS |
| $\mathbb{I}(\cdot)$ | indicator function | n_p | number of time-independent optimization variables |
| $\mathbb{N}_0^{n_x}$ | set of n_x -dimensional natural numbers with zero | n_s | number of state optimization variables |
| $\mathbb{P}[\cdot]$ | probability operator | n_t | number of time nodes |
| $\mathcal{N}(\cdot)$ | normal distribution | n_u | number of control optimization variables |
| \mathcal{R}_q | uncertainty set associated with $\tilde{\mathbf{q}}$ | n_x | number of uncertain basic quantities |
| μ_{\bullet} | mean value of $\tilde{\bullet}$ | $o(\cdot)$ | objective function |
| \bullet_{max} | upper bound on \bullet | o_{in} | inner-loop objective function |
| \bullet_{min} | lower bound on \bullet | PC | polynomial chaos degree |
| \mathbf{d} | vector of problem data | q_i | number of collocation nodes in i th dimension |
| $\phi_k(\tilde{x}_i)$ | univariate gPC basis functions | r_i | degree of the univariate gPC basis functions |
| $\Phi_m(\tilde{\mathbf{x}}_i)$ | n_x -variate gPC basis functions | s_f | scaling factor |
| σ_{\bullet} | standard deviation of $\tilde{\bullet}$ | t | time vector |
| $\tilde{\xi}(t)$ | uncertain state trajectory vector | t_0 | initial time |
| $\tilde{\mathbf{p}}$ | time-independent uncertain optimization variable vector | t_f | final time |
| $\tilde{\mathbf{p}}_p$ | uncertain plant optimization variable vector | t_{switch} | control switching time in the case study |
| $\tilde{\mathbf{q}}$ | column vector of uncertain quantities | v | objective function parameter in epigraph form |
| $\tilde{\mathbf{u}}(t)$ | uncertain control trajectory vector | w_n | normalization factor |
| $\tilde{\mathbf{x}}$ | vector of uncertain basic quantities | w_p | penalty weight |
| $\tilde{\bullet}$ | uncertain variable associated with \bullet | y_{pc} | PC th-degree gPC approximation of $\tilde{y}(\cdot)$ |
| $\tilde{\mathbf{d}}$ | uncertain problem data | Q | total number of collocation nodes |