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...
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]:

\[
\begin{align*}
\text{minimize:} & \quad o = \int_{t_0}^{t_f} \ell(t, u, \xi, p, d) dt + m(p, \xi_0, \xi_f, d) \\
\text{subject to:} & \quad g(t, u, \xi, p, \xi_0, \xi_f, d) \leq 0 \\
& \quad h(t, u, \xi, p, \xi_0, \xi_f, d) = 0 \\
& \quad \xi(t_0) = \xi_0, \quad \xi(t_f) = \xi_f, \quad u(t) = u, \quad \xi(t) = \xi \\
& \quad d(t) = d
\end{align*}
\]

(1a)

(1b)

(1c)

(1d)

(1e)

where \( t \in [t_0, t_f] \) is the time horizon, \( \{u, \xi, p\} \) are the collection of optimization variables including the open-loop control trajectories \( u(t) \in \mathbb{R}^n_u \), state trajectories \( \xi(t) \in \mathbb{R}^n_\xi \), and the vector of time-independent optimization variables \( p \in \mathbb{R}^n_p \), respectively. Note that \( p \) may entail plant optimization variables \( p_p \) and/or time-independent control optimization variables \( p_c \) [19, 22] (i.e., gains), such that \( p = [p_p, p_c] \). 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 \( g(\cdot) \) and \( h(\cdot) \), respectively. The transition or state derivative function \( f(\cdot) \) 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 \( d \in \mathbb{R}^n_d \).

In the remainder of this article, we assume that constraints associated with the initial and final conditions \( \{\xi_0, \xi_f\} \) are already included in \( h(\cdot) \) or \( g(\cdot) \). We will often drop the explicit dependence 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]:

\[
\begin{align*}
\text{minimize:} & \quad \mathbb{E}[\tilde{o}(t, \tilde{u}, \tilde{\xi}, \tilde{p}, \tilde{d})] \\
\text{subject to:} & \quad \mathbb{E}[\tilde{g}(t, \tilde{u}, \tilde{\xi}, \tilde{p}, \tilde{d})] \leq 0 \\
& \quad h(t, \tilde{u}, \tilde{\xi}, \tilde{p}, \tilde{d}) = 0 \\
& \quad \tilde{\xi}(t) - f(t, \tilde{u}, \tilde{\xi}, \tilde{p}, \tilde{d}) = 0 \\
& \quad \tilde{u}(t) = \tilde{u}, \quad \tilde{\xi}(t) = \tilde{\xi}, \quad \tilde{d}(t) = \tilde{d}
\end{align*}
\]

(2a)

(2b)

(2c)

(2d)

(2e)

where the expectation of \( \tilde{o}(\cdot) \), which is a function composition of \( o(\cdot) \), is optimized over the set of optimization variables \( \{\tilde{u}, \tilde{\xi}, \tilde{p}\} \), and is subject to the expectation of \( \tilde{g}(\cdot) \) (i.e., a function composition of \( g(\cdot) \)), analysis-type equality constraints \( h(\cdot) \), and uncertain dynamic system equality constraints in Eq. (2d). Note that \( \mathbb{E}[\tilde{o}(\cdot)] \) and \( \mathbb{E}[\tilde{g}(\cdot)] \) 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 \( \tilde{u}(t) \), uncertain state processes \( \tilde{\xi}(t) \), time-independent uncertain optimization variables \( \tilde{p} \), and time-dependent/time-independent uncertain problem data \( \tilde{d} \).

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 \( \{\tilde{u}, \tilde{\xi}, \tilde{p}\} \) 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 OLSC and OLMC structures and their effects on states of an arbitrary boundary-value UCCD problem.

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 OLSC or an OLMC structure. OLSC 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 (OLSC). 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, OLSC 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, OLSC 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 OLSC 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 OLSC 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 OLSC 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 OLSC-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 OLSC 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 OLSC uncertainty organization remains a valuable tool in constructing open-loop trajectories for a wide range of reference tracking applications.

In terms of implementation, OLSC 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 uncertainty \(^1\) 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 \(p_{\text{min}} \leq p \leq p_{\text{max}}\). 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:

\[
p_{\text{min}} + k_s \sigma_p \leq \tilde{p}_p \leq p_{\text{max}} - k_s \sigma_p
\]

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...
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 Ricatti 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 $\tilde{x} \in \mathbb{R}^{n_{u}}$ 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 $n_{u} = n_{p}$. Using $\tilde{x}$ 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:

$$
\hat{o}_{m} = \frac{1}{N_{mcs}} \sum_{j=1}^{N_{mcs}} o(t, u_{j}, \xi_{j}, p_{j}, d_{j})
$$

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

$$
\hat{o}_{\sigma}^{2} = \frac{1}{N_{mcs} - 1} \sum_{j=1}^{N_{mcs}} \left( o(t, u_{j}, \xi_{j}, p_{j}, d_{j}) - \hat{o}_{m} \right)^{2}
$$

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

$$
\hat{P}_{f,i} = \frac{1}{N_{mcs}} \sum_{j=1}^{N_{mcs}} \mathbb{I}(t, u_{j}, \xi_{j}, p_{j}, d_{j})
$$

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

$$
\mathbb{I}(t, u_{j}, \xi_{j}, p_{j}, d_{j}) = \begin{cases} 0 & \text{if } g_{i}(\cdot) \leq 0 \\ 1 & \text{if } g_{i}(\cdot) > 0 \end{cases}
$$

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 $P \ll 1$). The convergence rate of the expected value in MCS is $O(1/ \sqrt{N_{mcs}})$ [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 $\tilde{x}$ is a random vector with mutually independent components
and distribution $F_{\tilde{x}}(x) = \mathbb{P}(\tilde{x}_1 \leq x_1, \ldots, \tilde{x}_n \leq x_n)$. If the random elements in $\tilde{x}$ are not independent, then a Karhunen–Loève decomposition [68] or a Rosenblatt transformation [69] should first be applied. The mutual independence implies that $F_{\tilde{x}}(x) = \prod_{i=1}^{n} F_{\tilde{x}_i}(x_i)$. The univariate gPC basis functions of degree up to $r_i$ are denoted by $\{\phi_{i}(\tilde{x}_i)\}_{k=0}^{r_i}$ and, for continuous $\tilde{x}_i$, satisfy the following orthogonality condition:

$$\mathbb{E}\left[\phi_{m}(\tilde{x}_i)\phi_{n}(\tilde{x}_i)\right] = \int \phi_{m}(x)\phi_{n}(x)dF_{\tilde{x}_i}(x) = \delta_{mn} \gamma_m$$

where $0 \leq m, n \leq r_i$, $\gamma_m = \mathbb{E}[\phi_{m}^2(\tilde{x}_i)]$ where $\delta$ is the Kronecker delta function and $\gamma$ is a normalization factor. The set of univariate orthogonal basis functions are simply obtained based on the probability distribution of $\tilde{x}_i$. 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 $\{\phi_{i}(\tilde{x}_i)\}_{k=0}^{r_i} = \{H_0(\tilde{x}_i), H_1(\tilde{x}_i), \ldots, H_{r_i}(\tilde{x}_i)\}$, where $H(\cdot)$ refers to Hermite polynomial basis. Note that if $\tilde{x}_i$ 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 $\{\phi_{i}(\tilde{x}_i)\}_{k=0}^{r_i}$ for $i = 1, \ldots, n_x$ can then be used to construct the $n_x$-variate gPC basis functions $\Phi_m(\tilde{x})$. The resulting polynomials space can be defined as:

$$\Phi_m(x) = \gamma_m \prod_{i=1}^{n_x} \phi_{m_i}(\tilde{x}_i)$$

where $\gamma_m$ is the normalization factors and $\gamma_m = \mathbb{E}[\phi_{m}^2(\tilde{x}_i)]$ where $\delta$ is the Kronecker delta function and $\gamma$ is a normalization factor. The set of univariate orthogonal basis functions are simply obtained based on the probability distribution of $\tilde{x}_i$. 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 $\{\phi_{i}(\tilde{x}_i)\}_{k=0}^{r_i} = \{H_0(\tilde{x}_i), H_1(\tilde{x}_i), \ldots, H_{r_i}(\tilde{x}_i)\}$, where $H(\cdot)$ refers to Hermite polynomial basis. Note that if $\tilde{x}_i$ 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].

The resulting polynomials space can be defined as:

$$\Phi_m(x) = \gamma_m \prod_{i=1}^{n_x} \phi_{m_i}(\tilde{x}_i)$$

where $\gamma_m = \mathbb{E}[\phi_{m}^2(\tilde{x}_i)]$ where $\delta$ is the Kronecker delta function and $\gamma$ is a normalization factor. The set of univariate orthogonal basis functions are simply obtained based on the probability distribution of $\tilde{x}_i$. 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 $\{\phi_{i}(\tilde{x}_i)\}_{k=0}^{r_i} = \{H_0(\tilde{x}_i), H_1(\tilde{x}_i), \ldots, H_{r_i}(\tilde{x}_i)\}$, where $H(\cdot)$ refers to Hermite polynomial basis. Note that if $\tilde{x}_i$ 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].

From here, we can use either a Galerkin or a collocation formulation of gPC to calculate the unknown coefficients $\gamma_m(t)$ 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 mth unknown coefficient, $\hat{y}_m(t)$ can be obtained using Eq. (11):

$$\hat{y}_m(t) = \mathbb{E}\left[y(t, x)\Phi_j(\tilde{x})\right] = \int y(t, x)\Phi_j(\tilde{x})dF_{\tilde{x}}(x)$$

where $\Gamma$ 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 $\alpha_q$, such that $(x_j, \alpha_q)^{0}_{j=1}$, where $Q = \otimes_{i=1}^{n_x} q_i$. 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
the accuracy of integration. Using the quadrature rule, Eq. (13) can be written as:

\[ \hat{y}_m(t) = \sum_{j=1}^{Q} y(t, x_j) \Phi_m(x_j) \alpha_{w_j} \]  

(14)

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

\[ \mathbb{E}[\tilde{y}(t, \bar{x})] = \mathbb{E}[\tilde{y}(t, \bar{x})] = \hat{y}_1(t) \]

\[ \sigma^2_{\tilde{y}(t, \bar{x})} = \sigma^2_{\tilde{y}(t, \bar{x})} = \sum_{m=2}^{M} \sigma^2_{m}(t) \]  

(15)

Figure 2 illustrates all of the steps involved in constructing gPC basis functions for an arbitrary, two-dimensional uncertain problem.

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.

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.

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:

\[ \begin{aligned}
& \text{minimize:} \quad -\xi_1(t_f) \\
& \text{subject to:} \quad u - u_{\text{max}} \leq 0 \\
& \quad u_{\text{min}} - u \leq 0 \\
& \quad \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 1 - k \\ 0 \end{bmatrix} \\
& \quad \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
& \quad \xi_1(t_0) = 0, \quad \xi_2(t_f) = 0 \\
& \quad \text{where:} \quad u(t) = u, \quad \xi(t) = \xi 
\end{aligned} \]  

(16a)

(16b)

(16c)

(16d)

(16e)

(16f)

In this problem, \( u \) is the open-loop control moment applied to the solar array, \( \xi_1 \) and \( \xi_2 \) 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 \( u_{\text{min}} \) and \( u_{\text{max}} \), 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 \( \bar{k} \), uncertain problem data \( J \), and uncertain initial boundary condition for the second state variable \( \xi_2(t_f) \). 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 objective function and inequality constraints deterministic CCD problem may be formulated as SE-UCCD. In

4.1 Stochastic in Expectation (SE-UCCD)

implementation of this formulation. Therefore, the simple SASA problem readily lends itself to the OLMC structure.

In the outer-loop WCR-UCCD problem formulation, the OLMC structure solves the problem subject to all boundary conditions associated with e
czh side and geometry of the uncertainty set. We assume that uncer-
tainties belong to the following uncertainty set:

\[ \tilde{\xi}_t = \begin{bmatrix} \tilde{\xi}_1 \\ \tilde{\xi}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ -k \\ 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad \text{(a.s.)} \]  

\[ \tilde{\xi}_t(t_f) = \tilde{\xi}(t) = 0 \quad \text{(if OLMC)} \]

where: \( k = N(\mu_k, \sigma_k) \), \( \tilde{J} = N(\mu_J, \sigma_J) \)

\[ u(t) = u, \tilde{\xi}(t) = \tilde{\xi} \]

In this equation, \( N(\cdot) \) describes a normal distribution, \( \mu_k, \mu_J, \) and \( \mu_{\xi_2} \) are the mean values of the solar array stiffness, inertia ratio, and the initial condition for \( \xi_2 \), respectively. Equation (17c) is included to ensure the feasibility of any boundary solution according to the discussion in Sec. (2.2). In this equation, \( k \) 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:

\[ \mathcal{R}_q := \left\{ \tilde{q} \mid \| \delta_q \| \leq 1 \right\} \quad \text{where} \quad \delta_q = \tilde{q} - \hat{q} \]  

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

\[ \min \quad u_{d_k} \]

subject to:

\[ v - \Phi(t, u, \tilde{\xi}, \tilde{\xi}_{2,b}) \leq 0 \]  

Eqs. (16b)–(16c)  

\[ k_o \sigma_k - \mu_k \leq 0 \]  

where: \( u(t) = u \)

where \( u(t) \) and \( \mu_k \) 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:

\[
\begin{align*}
\text{minimize:} & \quad o_m = \xi_1(t_f) \\
\text{subject to:} & \quad \text{Eq. (16d)}
\end{align*}
\]  

(20a)

\[
\xi(t_0) = 0
\]  

(20b)

\[-0.3 \leq \xi_2(t_f) \leq 0.3
\]  

(20c)

where:  

\[
k(\mu_k) \in \mathcal{R}_k, \quad J(\mu_j) \in \mathcal{R}_j
\]  

(20d)

\[
\xi_2(\mu_{\xi_2(t_0)}) \in \mathcal{R}_{\xi_2(t_0)}, \quad \xi(t) = \xi
\]  

(20e)

(20f)

In this equation, the number of optimization variables include \(n_r = 2\), and \(n_p = 3\). Note that it is more natural for the WCR-UCCD formulation to be implemented through an OLSC 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 OLSC 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 formulation was solved using MCS and gPC. All of the results and comparison 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 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 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, \(\delta\) is the objective function associated with the corresponding formulation introduced in Eqs. (17a) and (19a), respectively.

<table>
<thead>
<tr>
<th>Category</th>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>defects</td>
<td>trapezoidal (TR)</td>
</tr>
<tr>
<td></td>
<td>mesh</td>
<td>equidistant</td>
</tr>
<tr>
<td></td>
<td>composite</td>
<td>TR</td>
</tr>
<tr>
<td></td>
<td>solver</td>
<td>fmincon</td>
</tr>
<tr>
<td></td>
<td>tolerance</td>
<td>(10^{-6})</td>
</tr>
<tr>
<td>SE-UCCD</td>
<td>(N_{mcs})</td>
<td>10,000</td>
</tr>
<tr>
<td></td>
<td>(Q)</td>
<td>(10^3)</td>
</tr>
<tr>
<td></td>
<td>(r_f)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>(M)</td>
<td>(9^3)</td>
</tr>
<tr>
<td>WCR-UCCD</td>
<td>derivatives</td>
<td>fmincon</td>
</tr>
<tr>
<td></td>
<td>(n_t)</td>
<td>100</td>
</tr>
</tbody>
</table>

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, \(\delta\) 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.

<table>
<thead>
<tr>
<th>Form.</th>
<th>Str.</th>
<th>$\bar{\theta}$</th>
<th>$\mu_\kappa$</th>
<th>$t(s)$</th>
<th>$t_{\text{switch}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCD</td>
<td>OLSC</td>
<td>-0.301</td>
<td>3.441</td>
<td>3</td>
<td>0.727</td>
</tr>
<tr>
<td>Stc-MCS</td>
<td>OLMC</td>
<td>-0.308</td>
<td>3.311</td>
<td>5717</td>
<td>0.737</td>
</tr>
<tr>
<td>Stc-gPC</td>
<td>OLMC</td>
<td>-0.306</td>
<td>3.185</td>
<td>562</td>
<td>0.742</td>
</tr>
<tr>
<td>$</td>
<td>\Delta</td>
<td>$</td>
<td>-</td>
<td>0.65%</td>
<td>3.81%</td>
</tr>
<tr>
<td>WcR</td>
<td>OLSC</td>
<td>0.204</td>
<td>0.705</td>
<td>2592</td>
<td>0.838</td>
</tr>
</tbody>
</table>

$^1$ 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 $N_{\text{mcs}} = 10,000$ number of samples for the MCS-based stochastic UCCD problem, the solution (which is described in Table 3) converged to the objective value of $\bar{\theta} = -0.308$, 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 $\bar{\theta}$ 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 $t_{\text{switch}}$ 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 4e, the expected value response for optimal state trajectories from both MCS and gPC-based methods are slightly higher than 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 $\pm \sigma_\kappa$ 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 $\mu_\kappa = 0.705$. 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, $\xi_{\text{wcr}}(t)$ is much lower than the deterministic case. Note, however, that $\xi_{\text{wcr}}(t_f)$ 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-
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 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 $\xi_2$ in common, but differ in $k$ dimension. Since $\mu_k$ 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 $(\mu_k \pm k_0 \sigma_k, -k_0 \sigma_{\xi_2}, k_0 \sigma_J)$ with $\xi_1(t_f) = 0.157$. According to Fig. 7c, $\xi_2$ trajectories associated with the worst-case follow a lower (slower) path compared to the determin-
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 OLSC 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 $\tilde{\sigma}_q = s_f \sigma_q$, where $\tilde{\sigma}_q$ 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, $\mu_k$ 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 $\mu_k$ 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 inclusion 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

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

\(\alpha_w\) quadrature weights
\(\bar{\omega}(\cdot)\) composite function of \(\omega(\cdot)\)
\(\xi(t)\) state trajectories
\(\xi_0\) initial conditions
\(\xi_f\) 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
˙ time-derivative of ⋅
ℓ(⋅) Lagrange term
Γ finite domain of distribution function
γ normalization factor
ˆ nominal quantities used to construct uncertainty sets
\( \hat{\mathcal{P}}_{f,i} \) estimate of the probability of failure for ith constraint
\( \hat{\mathbb{E}} \) unbiased estimate of the mean of \( \bar{\mathbb{E}} \)
\( \hat{\mathbb{E}}_\sigma \) unbiased estimate of the variance of \( \bar{\mathbb{E}} \)
\( \hat{m}(\cdot) \) gPC expansion coefficients
\( \mathbb{E}[-\cdot] \) expected value operator
\( \mathbb{I}(\cdot) \) indicator function
\( \mathbb{N}^n_0 \) set of \( n \times \)-dimensional natural numbers with zero
\( \mathbb{P}[-\cdot] \) probability operator
\( \mathcal{N}(\cdot) \) normal distribution
\( \mathcal{R}_q \) uncertainty set associated with \( \bar{q} \)
\( \mu_\cdot \) mean value of \( \bar{\cdot} \)
\( \sigma_{\cdot \max} \) upper bound on \( \bar{\cdot} \)
\( \sigma_{\cdot \min} \) lower bound on \( \bar{\cdot} \)
\( d \) vector of problem data
\( \phi_i(\bar{x}_i) \) univariate gPC basis functions
\( \Phi_m(\bar{x}_i) \) \( n \times \)-variate gPC basis functions
\( \sigma_\cdot \) standard deviation of \( \bar{\cdot} \)
\( \xi(t) \) uncertain state trajectory vector
\( \bar{p} \) time-independent uncertain optimization variable vector
\( \bar{p}_p \) uncertain plant optimization variable vector
\( \bar{q} \) column vector of uncertain quantities
\( \bar{u}(t) \) uncertain control trajectory vector
\( \bar{x} \) vector of uncertain basic quantities
\( \bar{\cdot} \) uncertain variable associated with \( \cdot \)
\( d \) uncertain problem data
\( \bar{y}(\cdot) \) an arbitrary second-order variable or process
\( F_{g}(x) \) joint probability distribution
\( H_k(\bar{x}_i) \) Hermite polynomial basis functions
\( J \) inertia ratio between the solar array and the bus
\( K \) solar array stiffness
\( k_s \) constraint shift index
\( m(\cdot) \) dimension of the \( n \times \)-variate polynomials
\( M \) last element in the support of some discrete distributions
\( n_d \) number of elements in the vector of problem data
\( n_{mcs} \) number of samples in MCS
\( n_p \) number of time-independent optimization variables
\( n_s \) number of state optimization variables
\( n_t \) number of time nodes
\( n_u \) number of control optimization variables
\( n_x \) number of uncertain basic quantities
\( o(\cdot) \) objective function
\( o_{in} \) inner-loop objective function
\( PC \) polynomial chaos degree
\( q_i \) number of collocation nodes in ith dimension
\( r_i \) degree of the univariate gPC basis functions
\( s_f \) scaling factor
\( t \) time vector
\( t_0 \) initial time
\( t_f \) final time
\( \tau_{\text{switch}} \) control switching time in the case study
\( v \) objective function parameter in epigraph form
\( w_n \) normalization factor
\( w_p \) penalty weight
\( \bar{y}_{pc} \) \( PC \)-th-degree gPC approximation of \( \bar{y}(\cdot) \)
\( Q \) total number of collocation nodes