CEAS-GNC-2024-057

EuroGNC CEAS Navigation Guidance Control Bristol, UK June 11th-13th University of BRISTOL ROYAL AERONAUTICAL SOCIETY AIAA 2024

A Scenario-based Approach to Robust Control Design

Luis G. Crespo Senior Research Scientist, Dynamic Systems and Controls Branch, NASA Langley Research Center, Hampton, VA, 23681, USA; luis.g.crespo@nasa.gov

Tanner Slagel Research Mathematician, Safety-Critical Avionics System Branch, NASA Langley Research Center, Hampton, VA, 23681, USA; joseph.t.slagel@nasa.gov

Sean P. Kenny Senior Research Scientist, Dynamic Systems and Controls Branch, NASA Langley Research Center, Hampton, VA, 23681; sean.p.kenny@nasa.gov

ABSTRACT

This paper proposes control tuning strategies for an ensemble of scenarios. Each of these scenarios might correspond to a different open-loop plant and/or an operating condition. Overfitting is prevented by ensuring that the closed-loop requirements are satisfied when the values prescribing such scenarios are perturbed. To this end we first model the perturbed scenarios as sample sets of finite size. Relaxed chance-constrained optimization is then used to seek controllers with varying degrees of robustness. For instance, we can deliberately eliminate a given number of scenarios from the dataset in order to obtain a riskier controller with a better performance, or we might seek a conservative controller that satisfies the closed-loop requirements with an acceptably high probability for as many perturbed scenarios as possible. The scenarios for which the requirements are not met, by either physics-based limitations or choice, are optimally chosen while the controller gains are searched for. The design of a feedback control system having a non-collocated sensor-actuator pair with time domain requirements is used for illustration.

Keywords: Data-driven control, chance-constrained optimization, uncertainty, robustness, outlier elimination.

1 Introduction

Consider a dynamical system whose performance depends on the decision variable $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ and the parameter $\delta \in \Delta \subset \mathbb{R}^{n_\delta}$, which might correspond to an uncertain parameter and/or an operating condition. This system is considered requirement compliant when $r_k(\theta, \delta) \leq 0$ for all $k = 1, \ldots n_r$, where $r_k : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\delta} \rightarrow \mathbb{R}$ is called a *Limit State Function* (LSF). These functions are assumed to be arbitrarily continuous, thereby making the framework applicable to settings in which they must be evaluated through simulation, e.g., $r_1(\theta, \delta) < 0$ is a settling time requirement for a non-linear closed-loop plant with uncertain parameter δ and controller's gains θ. Define the *success domain* of θ as

$$\mathcal{S}(\theta) = \bigcap_{k=1}^{n_r} \{\delta \in \Delta : r_k(\theta, \delta) \leq 0\}, \tag{1}$$

where each of the sets in the right hand side is an individual success domain. The complement of the success domain will be called the *failure domain*.

The field of optimization under uncertainty was pioneered in the 1950s by Dantzig and Charnes [1], who set the foundation for robust- and chanced-constrained-optimization respectively. Robust optimization seeks a design that optimizes the worst-case system's performance [2, 3], whereas chance-constrained optimization [4, 5] seeks a design that bounds or minimizes the *probability of failure*. The former approach prescribes the uncertainty in δ as a bounded and often convex set Δ, whereas the latter approach uses distributions or probability boxes.

In contrast to these approaches, the formulations proposed are driven by a finite number of realizations of δ called *scenarios* [6], and no attempt to characterize their underlying generating process is made. Hence, this approach eliminates upfront the need to prescribe a bounding set or a distribution along with the conservatism and subjectivity that such practice introduces to the resulting design. In particular, we seek an optimal design $\theta^\star$ that satisfies the requirements for the n realizations of δ in

$$\mathcal{D} = \left\{\delta^{(i)}\right\}_{i=1}^{n}. \tag{2}$$

Furthermore, we want to adjust the potentially high sensitivity of $\theta^\star$ to some of the *nominal scenarios* in (2). For instance, we might consider seeking the design point $\theta^\star$ that places as many scenarios onto the success domain as possible while minimizing the cost $J(\theta)$, or we might deliberately choose to let a few scenarios fall onto the failure domain in order to expand the feasible space and lower the cost. These scenarios, called outliers, will not be chosen empirically upfront but they will instead be chosen optimally during the search for $\theta^\star$. This practice mitigates the adverse effects that outliers often have on data-driven designs. The nominal scenarios should ideally be obtained experimentally through measurements. When experimental data is not available, they could be obtained synthetically through the simulation of an assumed probability measure.

Insufficient and inaccurately measured data threatens the relevance of a data-driven design because such a design might either be infeasible in practice or it might attain a cost far greater than the one predicted. An inaccurate metrology system, model-form uncertainty, measurement noise, and a small n make the characterization of δ imprecise, inaccurate or underdetermined. The extent by which these anomalies affect $\theta^\star$ will be mitigated by considering *perturbed scenarios*. As such, we want to prevent $\theta^\star$ from overfitting the elements of $\mathcal{D}$ by ensuring that the vicinity of each $\delta^{(i)}$ is also in $\mathcal{S}(\theta^\star)$. The robustness gains resulting from this practice often yield a loss in performance. The framework proposed enables trading off robustness and performance by ignoring subsets of each perturbed scenario not exceeding a maximum allowable probability of failure.

2 Perturbed Data

The pursuit of a robust design entails creating a model for the perturbed scenarios. These models, which should be set by the analyst according to the application at hand, must contain the unknown true value(s) of $\delta^{(i)}$ without being overly large. Otherwise they will lead to conservative designs that will underperform in practice.

One model class is given by compact sets. Denote as $\delta_s^{(i)} = \{\delta : \|\delta - \delta^{(i)}\| \leq \ell^{(i)}\}$ the set in the uncertain parameter space corresponding to the norm $\|\cdot\|$ for $\ell^{(i)} \geq 0$. This set leads to the sequence

$$\mathcal{D}_s = \left\{\delta_s^{(i)}\right\}_{i=1}^{n}. \tag{3}$$

Another model class is given by distributions. Denote $\delta_d^{(i)} = f_{\delta^{(i)}}(\delta^{(i)})$ as the joint density of the ith perturbed scenario having $\delta_s^{(i)}$ as the support set. These distributions lead to the sequence

$$\mathcal{D}_d = \left\{\delta_d^{(i)}\right\}_{i=1}^{n}. \tag{4}$$

As such there exists a known probability measure for each perturbed scenario in $\mathcal{D}_d$. The elements of (3) and (4) can be set according to the accuracy of the metrology system (when the scenarios were measured by system identification techniques with quantified measurement error), or according to expert opinion.

The design point θ can be learnt from either the nominal scenarios in $\mathcal{D}$ or from the perturbed scenarios in $\mathcal{D}_s$ or $\mathcal{D}_d$. The designs resulting from the former setting will be called *nominal* whereas those resulting from the later will be called *robust*. As expected, nominal designs might not exhibit the desired level of robustness, e.g., most perturbed sets might not be fully contained in $\mathcal{S}(\theta^\star)$ even though $\mathcal{D}$ is.

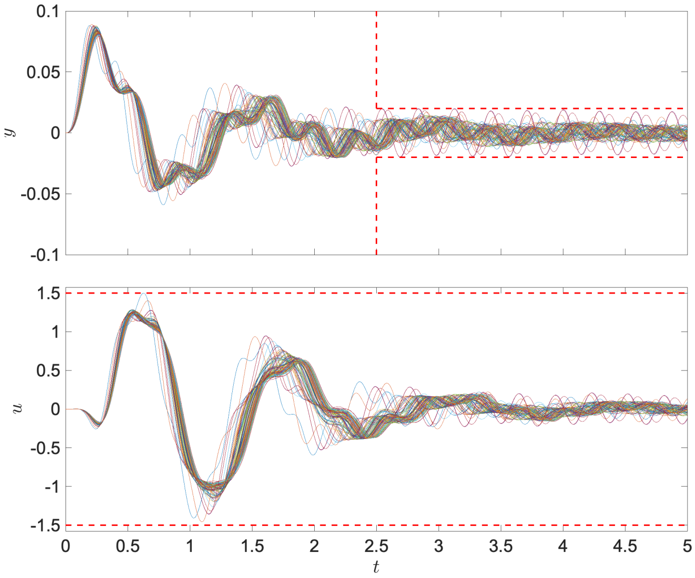

Fig. 1 Closed-loop $y(t)$ and $u(t)$ for the $n = 100$ scenarios scenarios in $\mathcal{D}$ and $\theta_1^\star$. The range limits are shown as dashed lines.

Robust models can be learned by using different strategies. The strategies below makes use of a multi-point representation of the uncertainty models in (3) and (4). Denote $\delta_p^{(i)} = \{\delta^{(i,j)}\}_{j=1}^m$ as a collection of m parameter points scattered in $\delta_s^{(i)}$ according to a predefined pattern. In the case of (3), these points might cover uniformly either the volume of $\delta_s^{(i)}$ or its surface; whereas in the case of (4), they might be quasi-random samples drawn from $\delta_d^{(i)}$. Hereafter we assume that one of such m points is $\delta^{(i)}$. These collections of points lead to the multi-point sequence

$$\mathcal{D}_p(m) = \left\{\delta_p^{(i)}\right\}_{i=1}^n . \tag{5}$$

The learning approaches developed below are based on (5) with a finite m. This practice might introduce an error, e.g., $\delta_p^{(i)} \subset \mathcal{S}(\theta)$ but[1] $\delta_s^{(i)} \not\subset \mathcal{S}(\theta)$, or not. Semi-infinite programming approaches exempt from this error are inapplicable to general LSFs. Besides, when they are applicable, e.g., the LSFs are polynomial and the perturbed sets are semi-algebraic, their high computational demands limit their usage to moderately large problems [7]. Worst-case and chance-constrained formulations to robust design based on the multi-point data sequence (5) are presented next. The elements of (3), (4) and (5) will be referred to as perturbed scenarios.

[1] This error can be reduced by sequentially appending to $\delta_p^{(i)}$ parameter points where previously found designs violate a requirement.

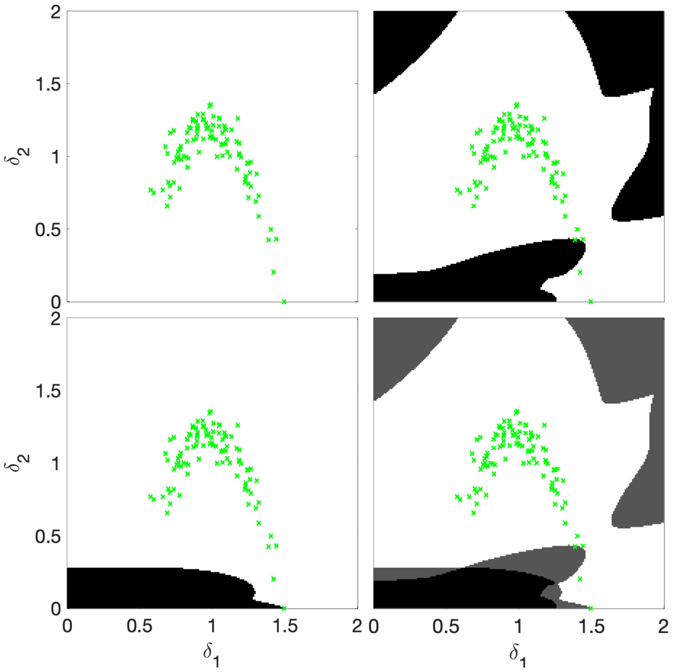

Fig. 2 Success and failure domains for $\theta_1^\star$. The top-left, top-right and bottom-left subplots show individual success (white) and failure (black) domains. The bottom-right subplot shows the success (white) and failure (non-white, where the darker the color the greater the number of requirement violations) domains. The scenarios falling into the success domain are marked with a green "×".

3 Worst-Case Design Framework

Consider the extension to the optimization program in [6] given by

$$\min_{\theta\in\Theta,\, \xi_i\geq 0} \quad c\left(\theta, \xi, \mathcal{D}_p\right) + \rho \sum_{i=1}^{n} \xi_i \tag{6}$$
$$\text{subject to :} \quad r_k\left(\theta, \delta^{(i,j)}\right) \leq \xi_i \quad \text{for } \; i = 1, \ldots n, \;\; j = 1, \ldots m, \;\; k = 1, \ldots n_r,$$

where c is the cost, $\rho \geq 0$ is a penalty coefficient, r_k is a LSF, and $\delta^{(i,j)}$ is an element of $\delta_p^{(i)}$ in (5). The solution to this program will be denoted as $\theta^\star$ and $\xi_i^\star$. The interpretation of (6) is that depending on the value of ρ some data points are allowed to fall onto the failure domain for the purpose of lowering c, but this action has itself a cost as expressed by the auxiliary optimization variables ξ_i: if $\xi_i > 0$ the constraint $r_k(\theta, \delta) \leq 0$ is relaxed to $r_k(\theta, \delta) \leq \xi_i$ in exchange for a cost increase of $\rho\,\xi_i$. Hence, (6) is a flexible scheme that enables the analyst to explore various solutions as ρ is varied from zero (no regret for constraints violation) to infinite (infinite regret for constraints violation). The formulation (6) is worst-case because $\xi_i^\star$ depends on the element of $\delta_p^{(i)}$ at which the LSF takes on the greatest value.

The perturbed scenarios whose multi-point sequence has at least one parameter point in the failure domain will be called *outliers*. The outliers, given by n_o indices in $\{1, \ldots n\}$, depend on the chosen value for ρ. Only the outliers contribute to the summation by taking on values proportional to the extent of the requirement violation, thereby making (6) for $n_o > 0$ a risk-based approach [8]. Program (6) might not have a solution for which all ξ_i are zero thereby making relaxation crucial. Relaxation enables the analyst to pursue the most robust design allowed by the assumed design architecture regardless of its ability to satisfy the requirements for all the training data, e.g., designing the PID controller that meets the requirements for as many uncertain plants as possible. The computational complexity of (6), which is independent of the number of uncertain parameters n_δ, is driven by the number of constraints $n \times m$.

Example 1: Consider the design of a feedback controller for an uncertain plant with three degrees of freedom having a non-collocated sensor-actuator pair. The closed-loop requirements defining (1) are asymptotic stability, and the system response $y(t, \delta, \theta)$ and the control signal $u(t, \delta, \theta)$ not exceeding range limits to an impulse disturbance. The latter two requirements define conflicting design objectives because fast system responses entail a greater control effort. In this setting $\theta \in \mathbb{R}^9$ are the controller's gains, whereas $\delta \in \mathbb{R}^{n_\delta}$ are the uncertain parameters of the plant[2]. Specifics on the control system are omitted due to space limitations. The instantiation of (6) used for control design is based on $n = 100$ uncertain plants, $n_r = 3$ LSFs, $n_\delta = 2$ uncertain parameters (so the results can be visualized), and the cost function[3]

$$c(\theta, \xi, \mathcal{D}_p) = \max_{\substack{i=1,\ldots n \\ j=1,\ldots m}} e^{-\beta \xi_i} \left\| y\left(t, \delta^{(i,j)}, \theta\right) \right\|_2, \tag{7}$$

where $\beta > 0$ is a parameter. The optimal cost, $c^\star = c(\theta^\star, \xi^\star, \mathcal{D}_p)$, upper bounds the norm of the response for the non-outlying elements of $\mathcal{D}_p$. This setting makes (6) a non-convex optimization program having constraints whose evaluation requires performing time simulations.

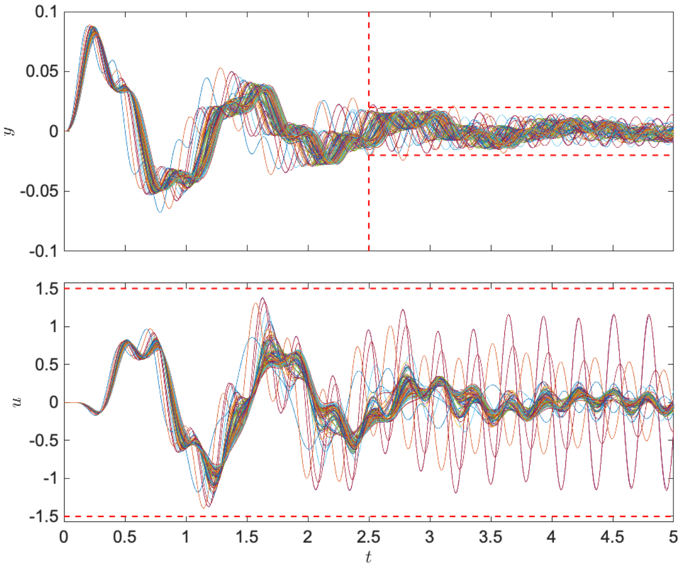

Fig. 3 Closed-loop response $y(t)$ and control input $u(t)$ for the $n = 100$ scenarios in $\mathcal{D}$ and $\theta_2^\star$.

We first synthesize two controllers for the nominal plants in $\mathcal{D}$, so $m = 1$ in (6). The first controller is designed by taking a large ρ value so as many data points as possible are placed in the success domain. This controller, with gains $\theta_1^\star$ and $\rho_1 = 0.1$, meets the requirements for all the plants. Fig. 1 shows the corresponding response and control signals along with the range limits, whereas Fig. 2 shows the individual and overall failure domains along the scenarios in $\mathcal{D}$. This figure was computed by evaluating the closed-loop response over a dense grid of parameter points. Note that all 100 scenarios are in the success domain but the controller is not robust (an arbitrarily small perturbation of the data point corresponding to the active constraint will lead to a requirement violation). A second controller was designed using $\rho_2 = 1 \times 10^{-6}$. This controller with gains $\theta_2^\star$ yields Figures 3 and 4. While all time responses for $\theta_1^\star$ remain within the limits, those for $\theta_2^\star$ violate the settling time requirement for four scenarios. These scenarios, chosen as outliers by the optimization, are hard to identified intuitively when the success domain is non-convex. The relaxation of the requirements for four outliers makes the volume of $\mathcal{S}(\theta_2^\star)$ smaller than that of $\mathcal{S}(\theta_1^\star)$ while reducing the cost from $c_1^\star = 3.334 \times 10^{-4}$ to $c_2^\star = 1.723 \times 10^{-4}$. As such, neglecting 4 outliers rendered a cost reduction of 48%.

[2] In contrast to standard data-driven control, the data herein correspond to different plant models rather than to input and output signals. This setting, which decouples model identification from control design, makes the approach amenable to model-based control.

[3] Note that the combination $\rho \ll 1$ and $\alpha > 0$ leads to the trivial solution $c^\star \approx 0$. As such, use $\alpha = 0$.

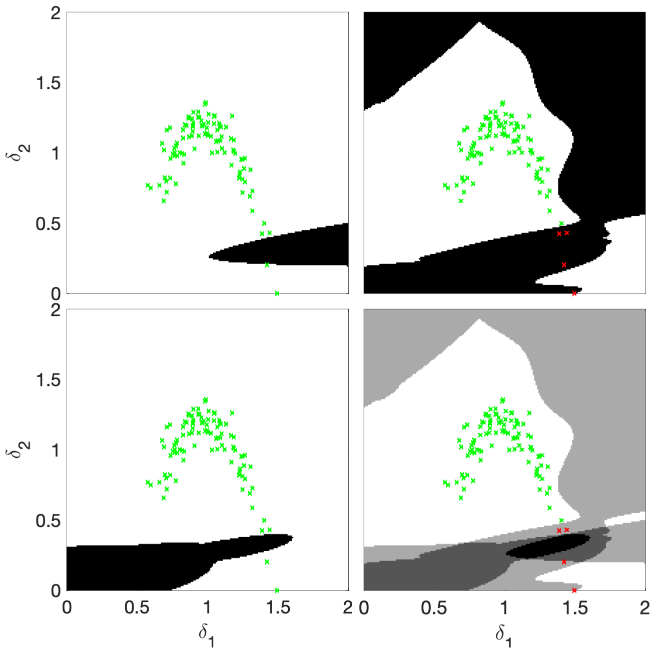

Fig. 4 Success and failure domains for $\theta_2^\star$. The top-left, top-right and bottom-left subplots show individual success (white) and failure (black) domains. The bottom-right subplot shows the success (white) and failure (non-white, where the darker the color the greater the number of requirement violations) domains. The scenarios falling into the success domain are marked with a green "×", otherwise they are marked with a red "×".

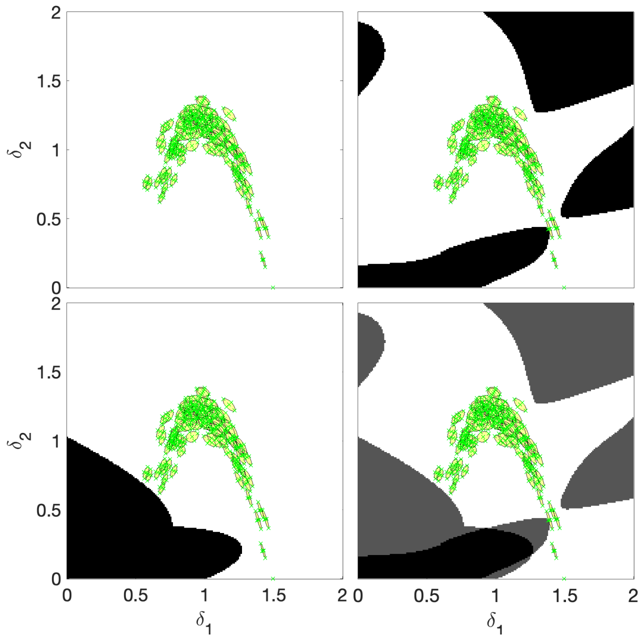

Fig. 5 Success and failure domains for $\theta_3^\star$. Previously defined conventions apply. The perturbed scenarios are the yellow ellipsoids whereas the elements of $\mathcal{D}_p$ are shown as a green "×".

Next we design a controller for the perturbed sets in (3) shown as ellipsoids in Fig. 5. To this end, we create the multi-point sequence $\mathcal{D}_p$ having the $m = 5$ elements shown. The value $\rho = 1 \times 10^3$ was chosen in order to place as many data points as possible inside the success domain. The corresponding controller, with gains $\theta_3^\star$, yields Figures 5 and 6. In contrast to $\theta_1^\star$ and $\theta_2^\star$, controller $\theta_3^\star$ satisfies the requirements for all the $n \times m$ parameter points in $\mathcal{D}_p$. This controller places all the perturbed scenarios in the success domain (as shown by an independent optimization-based analysis) even though only $m = 5$ points were used for design. A simpler control architecture or larger perturbations might prevent this outcome. Notice that the volume of $\mathcal{S}(\theta_1^\star)$ is greater than that of $\mathcal{S}(\theta_3^\star)$ but the robustness of the latter is greater because only $\theta_3^\star$ places all the perturbations inside the success domain. The notion of robustness enforced by (6) is tightly linked to the manner by which the training data in $\mathcal{D}_p$ spreads. As such, the resulting controller might be less robust elsewhere. This is exemplified in Figures 2 and 5 which show that the perturbed sets at the bottom-left and top-right are much closer to the failure domain of $\theta_3^\star$ than to the failure domain of $\theta_1^\star$. This controller achieves the cost $c_3^\star = 2.728 \times 10^{-4}$. A comprehensive analysis of the above controllers, which can be used to assess the effects of the sampling error caused by a finite number of points m when searching for θ, is presented below.

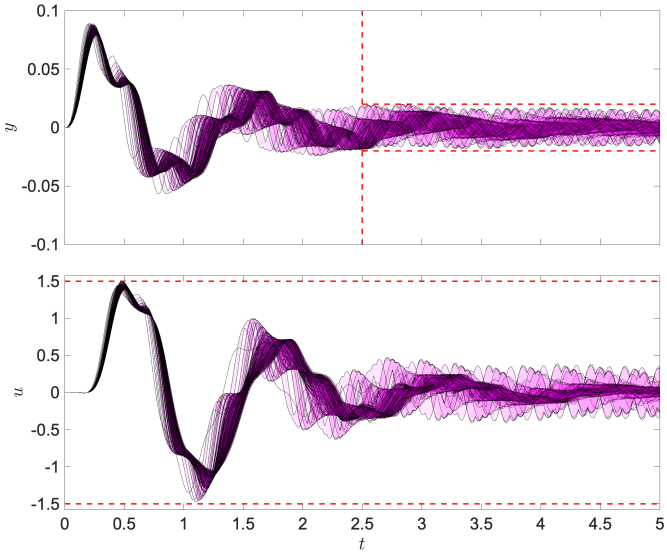

Fig. 6 Closed-loop response $y(t)$ and control input $u(t)$ for the $n = 100$ perturbed scenarios in $\mathcal{D}_p$ and $\theta_3^\star$. The interval-valued function enclosing all the time signals corresponding to the m elements of $\delta_p^{(i)}$ for all i are shown.

4 Chance-Constrained Design Framework

4.1 Preliminary developments

Approximations to the *Cumulative Distribution Function* (CDF) of a random variable and its inverse based on a finite number of observations are presented next. Consider the sequence $\mathcal{Z} = \{z_i\}_{i=1}^n$ that results from evaluating $z(p)$ at the points $\{p^{(i)}\}_{i=1}^n$ and sorting the resulting numbers so $z_i = z(p^{(j)})$ for some j and $z_i < z_{i+1}$. To make this dependency explicit we introduce the sorting operator $s(\cdot)$ such that $\mathcal{Z} = s(\{z(p^{(i)})\}_{i=1}^n)$. A continuous, piecewise linear approximation to the CDF of $z(p)$ is

$$F_{\mathcal{Z}}(z) \triangleq \begin{cases} 0 & \text{if } z \leq z_1 \\ \frac{1}{\sigma}\left(i - 1 + \frac{z - z_i}{z_{i+1} - z_i}\right) & \text{if } z_i < z \leq z_{i+1} \\ 1 & \text{otherwise,} \end{cases} \tag{8}$$

where $\sigma = n - 1$. The inverse CDF is approximated by

$$F_{\mathcal{Z}}^{-1}(\alpha) = \begin{cases} z_1 & \text{if } \alpha = 0, \\ z_i + \sigma(z_{i+1} - z_i)\left(\alpha - \frac{i-1}{\sigma}\right) & \text{if } 0 < \alpha < 1 \\ z_n & \text{otherwise,} \end{cases} \tag{9}$$

where $i = \text{argmin}_{j=1,\ldots n}\{\sigma\alpha - j + 1 : j - 1 \leq \alpha\sigma\}$ and $\alpha \in [0, 1]$. These approximations make standard gradient-based algorithms applicable to the forthcoming formulation.

4.2 Chance-constrained optimization

Note that it only takes a single element of $\delta_p^{(i)}$ falling into the failure domain for (6) to penalize the relaxation of a requirement. In fact, the greatest value taken by the limit state functions at the m sampling points prescribes the penalty term $\rho\xi_i$ in (6). In contrast, the developments below make the penalty term independent of the realizations of r_k below a threshold possibly greater than zero, thereby not penalizing requirement violations for some elements of $\delta_p^{(i)}$. As such, a design that satisfies the requirements for, say, 95% of all possible realizations of the distribution characterizing the perturbation incurs no penalty. Neglecting the worst-performing elements of each $\delta_p^{(i)}$ (those corresponding to the greatest cost) makes the resulting design riskier. This practice, however, might yield to sizable performance improvements, e.g., the value of the L2 norm in (12) for most $\delta^{(i,j)}$ points is lower, without impacting most of the data, e.g., the probability of failure corresponding to most distributions is not affected by such a threshold.

The approximation to the CDF in (8) along with its inverse allow solving chance-constrained programs. For instance, the chance-constrained extension of (6) is

$$\begin{aligned} \min_{\theta\in\Theta,\ \xi_i\geq 0} \quad & c\left(\theta, \xi, \mathcal{D}_p\right) + \rho \sum_{i=1}^{n} \xi_i \\ \text{subject to:} \quad & F_{\tilde{\mathcal{Z}}(\theta,\mathcal{D}_p,i,k)}(\xi_i) \geq \gamma_k \;\; \text{for} \;\; i = 1, \ldots n, \;\; k = 1, \ldots n_r, \end{aligned} \tag{10}$$

where c is the cost, $\rho \geq 0$ is the penalty coefficient,

$$\tilde{\mathcal{Z}}(\theta, \mathcal{D}_p, i, k) \triangleq s\left(\left\{r_k(\theta, \delta^{(i,j)})\right\}_{j=1}^{m}\right), \tag{11}$$

and $0 \ll \gamma_k < 1$ is the minimally acceptable probability of success for the k-th requirement. Hence, (10) seeks a design that minimizes the sum of c and a penalty term while bounding the individual success probabilities for most elements of (5) from below. This is a flexible scheme that enables the analyst to explore various solutions as ρ is varied from zero (no regret for having probabilities of success below γ_k) to infinite (infinite regret for having probabilities of success below γ_k). Large values of ρ will drive the ξ_i's near zero thereby ensuring that the individual success probabilities are no less than γ_k. Moderately large values of ρ will make some $\xi_i > 0$, therefore yielding a design for which some individual probabilities of success are below γ_k. The scenarios for which $\xi_i > 0$ will be called *outliers*. When $\gamma_k < 1$ a few points of $\delta_p^{(i)}$ might fall into the failure domain regardless of the corresponding scenario being an outlier or not. Note that evaluating the constraints of (10) by means of the empirical CDF make them piece-wise constant, thereby precluding the usage of standard gradient-based algorithms. Note that (10) is equivalent to (6) when $\gamma_k = 1$ for $k = 1, \ldots n_r$.

As in (6), the number of uncertain parameters is immaterial to (10) whereas the number of constraints $n \times m$ is the limiting factor. The empirical probabilities driving (10) might differ significantly from the actual probabilities when m is small. However, the computational cost of solving (10) with a large m, which entails evaluating the LSFs $m \times n$ times for each candidate design, makes this approach impractical. The non-zero terms in the summation quantify the extent by which the requirements are violated by the outliers thereby making (10) a risk-based approach[4]. This risk, which is driven by the chosen values for ρ and γ_k for $k = 1, \ldots n_r$, should be adjusted according to the importance of the underlying requirements

[4] Risk is commonly defined as the probability of an adverse outcome times the corresponding loss. In the setting above the adverse outcome is the violation of a requirement whereas ξ_i is a loss measure. Hence, the dependency of the penalty term on ξ makes $\theta^\star$ a *risk-based* design.

and the manner by which the data spreads, e.g., by making the radius $r^{(i)}$ in (3) depend on $\delta^{(i)}$. Worst-case designs might differ considerably from chance-constrained designs. Whereas (6) must drive all the points of most constellations in $\mathcal{D}_p$ to the success domain, (10) does not. This flexibility along with a soft notion of risk might render simpler design architectures preferable. For instance, a simple PID controller might be able to satisfy the requirements for 95% of the plants of each of the n distributions. Doing so for the remaining 5% might require adopting and managing a much more complex controller.

Example 2: Next we use (10) to synthesize controllers based on the same $n = 100$ uncertain plants, $n_r = 3$ LSFs, and $n_\delta = 2$ uncertain parameters used previously. The perturbed scenarios are modeled as normal distributions whose iso-likelihood set of 0.95 probability coincides with the sets shown in Fig. 5. The multi-point sequences required to compute the CDFs and their inverses are based on $m = 50$ points. Besides, we will use the cost function

$$\hat{c}(\theta, \xi, \mathcal{D}_p) = \max_{i=1,\ldots n} e^{-\beta\xi_i} F^{-1}_{\hat{\mathcal{Z}}_i}(\gamma), \tag{12}$$

where $\hat{\mathcal{Z}}_i \triangleq s(\{\|y(t, \delta^{(i,j)}, \theta)\|_2\}_{j=1}^{m})$ and $0 < \gamma \ll 1$. The optimal cost, $\hat{c}^\star = \hat{c}(\theta^\star, \xi^\star, \mathcal{D}_p)$, bounds the γ quantile of the L2 norm of the response for the non-outlying scenarios. Note that (12) and its gradient can be readily computed from (9).

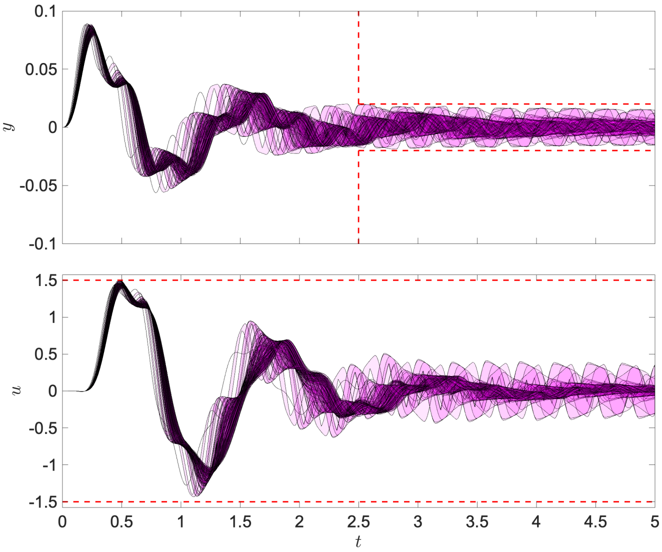

Fig. 7 Closed-loop responses $y(t)$ and $u(t)$ for the $n = 100$ scenarios in $\mathcal{D}_p$ and $\theta_4^\star$. Previously defined conventions apply.

A controller with gains $\theta_4^\star$ was first designed using $\rho = 1$. This controller, which yields Figs. 7 and 8, meets the chance-constraints for all $n = 100$ plants. Even though this controller did not render any outliers, a sample of a distribution violates the control limit requirement (see the red crosses in the bottom-left subplot). The empirical probability of violation for this scenario was about $1/m$. A Monte Carlo analysis with 5000 simulations shows that three distributions out of the $n = 100$ have failure probabilities below the 0.05 threshold. The comparison of Figures 5 and 8 shows that the failure domain for the chance-constrained design is closer to the perturbed scenarios than that for the worst-case controller. The optimal cost for this controller is $\hat{c}_4^\star = 2.698 \times 10^{-4}$, which is slightly lower than $\hat{c}_3^\star$. One last controller was designed by using the cost

$$\tilde{c}(\theta, \xi, \mathcal{D}_p) = \mathbb{E}\left[\, \|y(t, \delta_p^{(i)}, \theta)\|_2 \mid \xi_i = 0 \right], \tag{13}$$

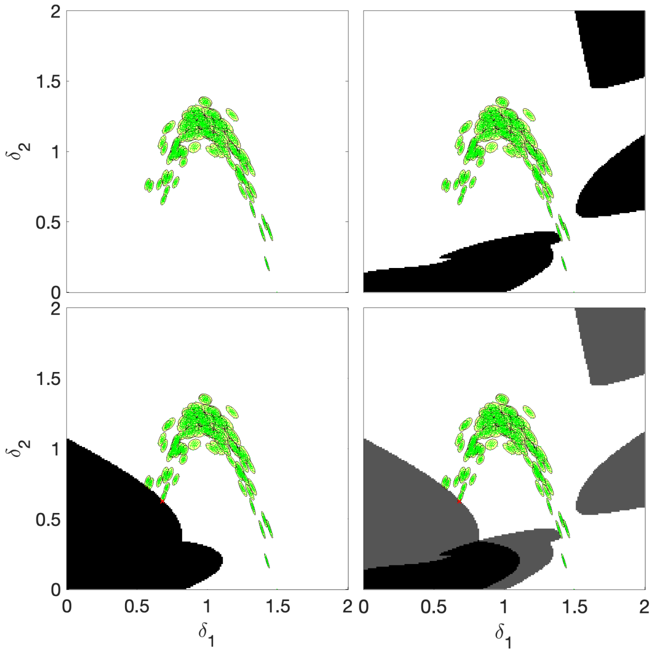

Fig. 8 Success and failure domains for $\theta_4^\star$. Previously defined conventions apply.

where $\mathbb{E}[\cdot|\cdot]$ is the conditional sample mean. Hence, this controller minimizes the mean of the norm of the $m(n-n_o)$ responses of the non-outlying scenarios. The parameter $\rho = 1$ led to a controller with no outliers. This controller, with gains $\theta_5^\star$ and cost $\tilde{c}_5^\star = 4.823 \times 10^{-5}$, yields Fig. 10. As before, samples of one distribution violate a requirement with an empirical probability of failure of $2/m$ (see red crosses in top-right subplot). However, in contrast to $\theta_4^\star$, the limiting requirement is the bounded response not the control input.

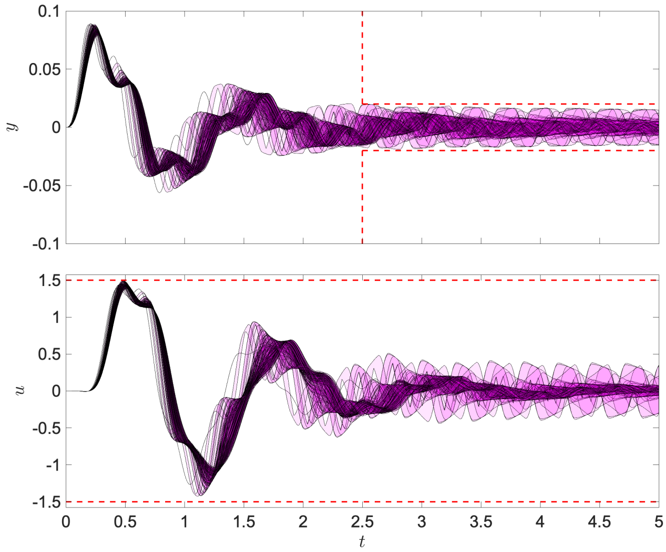

Fig. 9 Closed-loop responses $y(t)$ and $u(t)$ for the $n = 100$ scenarios in $\mathcal{D}_p$ and $\theta_5^\star$. Previously defined conventions apply.

A robustness analysis of the above controllers is presented next. To this end, a Monte Carlo analysis was carried out by evaluating the closed-loop performance for 4×10^5 simulations drawn from a Gaussian mixture of all distributions in (4). The CDFs of the Worst-case LSF and of the response are shown in Fig. 11. Recall that the probability of a requirement violation is given by one minus the value of the CDF at the top subplot evaluated at zero. These probabilities are very small for all controllers but for $\theta_4^\star$,

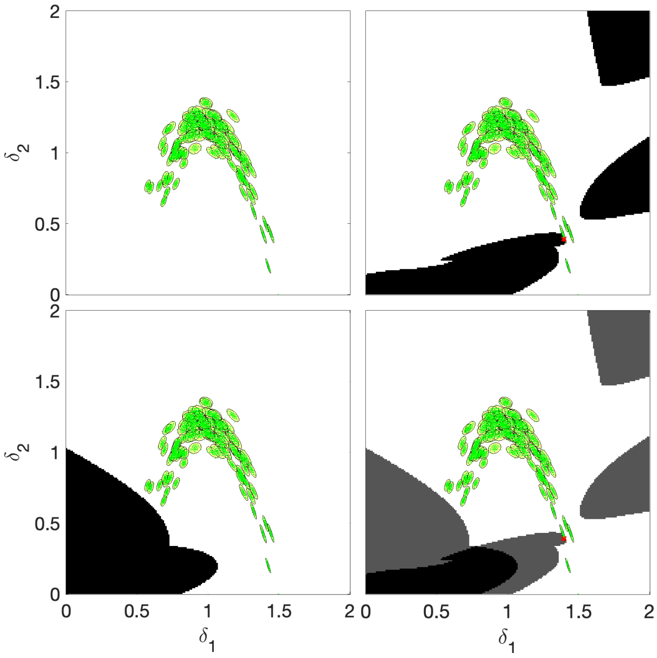

Fig. 10 Success and failure domains for $\theta_5^\star$. Previously defined conventions apply.

which violates the requirements with probability 0.05 intendedly. The error caused by assuming a finite m did not invalidate the desired robustness levels. The CDFs at the bottom subplot allow comparing the performance of various controllers. In this setting, the lower the quantile for a fixed value of the ordinate, the better the performance. For instance, $\theta_5^\star$ is the best performing controller when $F_{||y||_2} < 0.915$, whereas $\theta_2^\star$ is so for $F_{||y||_2} = 0.975$. Recall that the improved performance of $\theta_2^\star$ was traded-off against an increase in the failure probability. Unfortunately, this improvement does not extend to values of $F_{||y||_2}$ lower than 0.96. This is not the case for $\theta_4^\star$ and $\theta_5^\star$, which traded-off a sizable performance improvement (bottom plot) against a minor increase in the probability of failure (the CDFs cross zero above 0.999). Note that $\theta_4^\star$ and $\theta_5^\star$ exhibit a very similar performance (the means are 4.894×10^{-5} and 4.823×10^{-5}, and the 95 quantiles are 2.699×10^{-4} and 2.710×10^{-4} respectively) even though their worst-case LSF CDFs differ significantly.

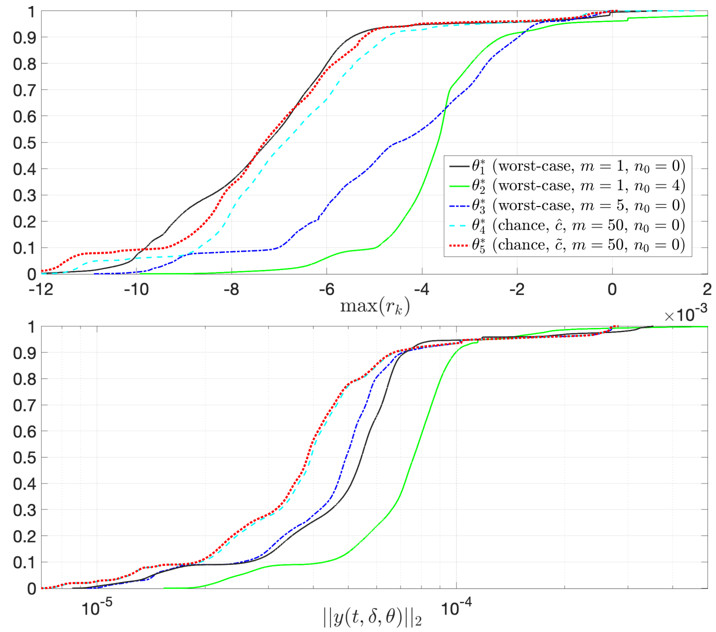

Fig. 11 CDFs of the worst-case LSF, $F_{\max(r_k)}$ (top), and of and $||y||_2$, $F_{||y||_2}$ (bottom), for several controllers.

5 Conclusion

This paper proposes a chance-constrained formulation for the design of robust data-driven controllers. Overfitting is prevented by seeking designs that satisfy the requirements at not only the values of the nominal scenarios but also at their vicinity. This framework enables the analyst to trade-off performance against robustness by eliminating the scenarios, either nominal or perturbed, that contribute most to degradation in closed-loop performance. These strategies can be naturally integrated to the Monte Carlo campaigns commonly used to assess control robustness, by bridging the goals of control design, typically evaluated using simplified LTI models, with those of control verification, typically evaluated using high-fidelity physics-based models.

References

[1] A. Charnes and W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

[2] J. Mayer and P. Kall. *Stochastic linear programing: Models, theory and Computation*. Springer Verlag, 2010.

[3] D. Bertsimas and A. Thiele. Robust and data-driven optimization: Modern decision making under uncertainty. *INFORMS*, 2014.

[4] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on stochastic programming: modeling and theory*. SIAM, Philadelphia, PA, 2009.

[5] D. Coit and E. Zio. The evolution of system reliability optimization. *Reliability Engineering and System Safety*, 192:106259, 2019.

[6] M. Campi, A. Care, and S. Garatti. The scenario approach: A tool at the service of data-driven decision making. *Annual Reviews in Control*, 52:1–17, 2021.

[7] M. Lacerda and L. Crespo. Interval predictor models for data with measurement uncertainty. In *American Control Conference*, pages 1487–1492, May 2017. DOI: 10.23919/ACC.2017.7963163.

[8] M. Chapman, M. Faub, and K. Smith. On optimizing the conditional value-at-risk of a maximum cost for risk-averse safety analysis. *IEEE Transactions on Automatic Control*, 2022. DOI: 10.1109/TAC.2022.3195381.