Scenario Optimization for MPC

Marco C. Campi, Simone Garatti, and Maria Prandini

1 Introduction

Model Predictive Control (MPC) is a methodology to determine control actions in the presence of constraints that has proven effective in many real applications. Instead of addressing an infinite-horizon problem, which would be hard to deal with due to computational difficulties, in MPC one solves at each point in time a finite-horizon constrained problem, and implements only the first control action that has been determined; then, the procedure is repeated at the next instant of time by shifting the prediction horizon ahead of one unit of time (receding horizon).

In many control problems, disturbances are a fundamental ingredient. In MPC, disturbances have been dealt with along two different approaches, namely robust MPC and stochastic MPC. In robust MPC (e.g., [4, 5, 23, 34, 47, 50]), the control cost is optimized against the worst disturbance realization, while also guaranteeing constraints satisfaction. The drawback with this approach is that it generates conservative control actions. To overcome this drawback, an average cost with probabilistic constraints is considered in stochastic MPC where a violation of the constraints is accepted provided the probability of this to happen is kept below a given threshold (e.g., [3, 9, 18, 19, 21, 46, 49, 53]). In the stochastic optimization literature, probabilistic constraints of this type are often called "chance-constraints," see, e.g., [44, 45].

M. C. Campi (✉)
Università di Brescia - Dipartimento di Ingegneria dell'Informazione, via Branze 38, 25123, Brescia, Italy
e-mail: marco.campi@unibs.it

S. Garatti · M. Prandini
Politecnico di Milano - Dipartimento di Elettronica, Informazione e Bioingegneria, piazza L. da Vinci 32, 20133, Milan, Italy
e-mail: simone.garatti@polimi.it; maria.prandini@polimi.it

S. V. Raković, W. S. Levine (eds.), *Handbook of Model Predictive Control*,
Control Engineering, https://doi.org/10.1007/978-3-319-77489-3_19

Chance-constraints are known for being very hard to deal with. One reason is that they are highly non-convex even when the original problem is born within a convex setup where the constraints are convex for any given realization of the disturbance. As a matter of fact, solutions to MPC with chance-constrained constraints have been proposed for specific cases only such as linear systems with either bounded and i.i.d. (independent and identically distributed) [9, 18, 19] or Gaussian [21] disturbances. In this chapter, we describe an alternative scheme to deal with stochastic MPC ([43]). This scheme is grounded in some recent developments in stochastic optimization where the chance-constraints are replaced by variants obtained by sampling finitely many realizations of the disturbance (scenario approach). Considering a finite sample of realizations makes the problem computationally tractable while the link to the original chance-constrained problem is established by a rigorous theory. With this approach, one gains the important advantage that no assumptions on the disturbance, such as boundedness, independence or Gaussianity, is required.

This chapter is organized as follows. In the first section the mathematical setup of study is introduced. After a digression to summarize some relevant results of the scenario approach in Section 3, Section 4 describes how the scenario methodology can be applied to MPC. The closing Section 5 presents a simulation study for a mechanical system.

2 Stochastic MPC and the Use of the Scenario Approach

Consider a linear system whose state $x_t \in \mathbb{R}^n$ evolves according to the equation

$$x_{t+1} = Ax_t + Bu_t + Dw_t,$$

where $u_t \in \mathbb{R}^m$ is the control input, $w_t \in \mathbb{R}^l$, $l \leq n$, is a stochastic disturbance with a possibly unbounded support, and D is full-rank.

We assume that the entire state vector of the system is known at each time instant and focus on the finite-horizon optimization problem that needs to be solved at each point in time τ of a stochastic MPC scheme that implements a receding horizon strategy. Specifically, we consider the following quadratic cost

$$J = \mathbb{E}\left[\sum_{i=1}^{M} x_{\tau+i}^T Q_i x_{\tau+i} + \sum_{i=0}^{M-1} u_{\tau+i}^T R_i u_{\tau+i}\right], \tag{1}$$

where M is the horizon length and $Q_i = Q_i^T \succeq 0$ and $R_i = R_i^T \succ 0$ have appropriate dimensions, subject to the probabilistic constraint:

$$\mathbb{P}\{f(x_{\tau+1}, \ldots, x_{\tau+M}, u_\tau, \ldots, u_{\tau+M-1}) \leq 0\} \geq 1 - \varepsilon, \tag{2}$$

where $f : \mathbb{R}^{n\times M+m\times M} \to \mathbb{R}^q$ is a q-dimensional function and the inner inequality in (2) is interpreted componentwise. In the above expressions, probability $\mathbb{P}$ refers to the stochastic nature of the disturbance and $\mathbb{E}$ is the expected value associated with it. In the probabilistic constraint (2), condition $f(x_{\tau+1},\dots,x_{\tau+M},u_\tau,\dots,u_{\tau+M-1}) \le 0$ is not required to hold for all possible disturbance realizations and parameter $\varepsilon \in (0,1)$ quantifies the admissible probability of constraint violation. Allowing for an ε violation improves the control system performance and, moreover, when the disturbance has unbounded support, allowing for a small probability of constraint violation can be the only way to avoid infeasibility of the optimization problem. In practice, applications exist where violating a constraint may result in severe damages of equipments or in important malfunctionings, in which case one may not be willing to allow for an ε-violation. In many other cases, however, sporadic constraint violations are tolerable and cause little damage. For example, exceeding the load capacity in power lines for a short time does not cause any plant damages and, in a totally different field, high blood glucose is not a cause of cellular damage if it happens for short periods. Similar examples can be found in a variety of contexts. This is the frame where stochastic MPC finds application.

In many cases, function f in (2) is used to enforce input saturation constraints in addition to constraints on the allowed state values. If, for instance, f is given by

$$f(x_{\tau+1},\dots,x_{\tau+M},u_\tau,\dots,u_{\tau+M-1}) = \begin{bmatrix} \sup\limits_{i=0,\dots,M-1} \|Su_{\tau+i}\|_\infty - \bar{u} \\ \sup\limits_{i=1,\dots,M} \|Cx_{\tau+i}\|_\infty - \bar{y} \end{bmatrix}, \tag{3}$$

where S and C are matrixes in $R^{q\times m}$ and $R^{p\times n}$ respectively, then, $\bar{u}$ and $\bar{y}$ are limits on linear combinations of the inputs and of the state values. In the following, we shall consider generic but convex functions f.

Note that when the noise is Gaussian and constraints are missing, minimizing (1) gives a standard LQG control problem which admits analytical solution. Instead, in the presence of constraints, or when the noise is not Gaussian, the problem of finding the optimal solution becomes quite challenging. Hence, one can concentrate on specific structures by which the control actions are determined.

To find a suitable structure, one can think of reconstructing the noise from the state according to the equation

$$w_{\tau+i} = D^\dagger(x_{\tau+i+1} - Ax_{\tau+i} - Bu_{\tau+i}),$$

where $D^\dagger$ is pseudo-inverse and then parameterize the control action as an affine function of the disturbance

$$u_{\tau+i} = \gamma_i + \sum_{j=0}^{i-1} \theta_{i,j} w_{\tau+j}, \tag{4}$$

with $\gamma_i \in \mathbb{R}^m$ and $\theta_{i,j} \in \mathbb{R}^{m\times n}$.[1] This parametrization was indeed proposed in [30] (and, independently in [6]) where it was also shown that (4) is equivalent to considering policies that are affine in the state (i.e., to every affine in the state policy $\mu_{\tau+i}$, there correspond γ_i and $\theta_{i,j}$ such that (4) returns the same control action as $\mu_{\tau+i}$ and vice versa).

The fundamental advantage gained by adopting (4) is that the control cost and the constraints become convex in the variables γ_i and $\theta_{i,j}$ (when the control action is parameterized as an affine function of the state, this fails to be true). We shall write this control cost and the constraints explicitly in Section 4 after introducing suitable notations. There, we shall further show that by sampling the noise realizations (scenario approach) and enforcing the constraints only on the realizations that have been sampled, one obtains a standard convex problem that can be solved with conventional optimization methods. The so-found solution carries precise guarantees of satisfaction of the original chance-constrained constraint. Proving this deep result calls for the use of the scenario theory that is briefly summarized in the next section.

3 Fundamentals of Scenario Optimization

Consider the following constrained convex optimization problem

$$\begin{aligned} &\min_{x\in\mathscr{X}\subseteq\mathbb{R}^d} \ell(x) \quad \text{subject to:} \\ &x \in \mathscr{X}_\delta, \quad \delta \in \Delta, \end{aligned} \tag{5}$$

where $\ell(x)$ is a convex function, $\delta \in \Delta$ is an uncertain parameter, and $\mathscr{X}$ and $\mathscr{X}_\delta$ are convex and closed sets. In normal situations, Δ has infinite cardinality. Uncertainty in (5) can be dealt with along two distinct approaches. The first one consists in enforcing satisfaction of all constraints, that is one optimizes the cost $\ell(x)$ over the set $\bigcap_{\delta\in\Delta}\mathscr{X}_\delta$ (robust approach). Alternatively, one may want to satisfy the constraints with "high probability" (stochastic approach). Along this second approach one sees the uncertainty parameter δ as a random element with a probability $\mathbb{P}$, and seeks a solution that violates at most a fraction of the constraints that has small probability (chance-constrained solution). This second approach is often more advantageous in that it returns less conservative designs.

Notoriously, finding a solution to (5) that carries a high probability of constraints satisfaction is a very difficult task [44]. In [7, 8], the following scenario problem is introduced, where N values of δ, say $\delta^{(1)},\dots,\delta^{(N)}$, are randomly sampled from $\mathbb{P}$ one independently of the others and these N values provide the only constraints that are enforced in the optimization problem:

---

[1] Often, the total number of parameters is reduced as compared to (4) by imposing internal relation among parameters. This is further discussed in Section 4. When all $\theta_{i,j}$ are set to zero, one obtains a classical setup where optimization is directly performed on the control actions.

$$\min_{x \in \mathscr{X} \subseteq \mathbb{R}^d} \ell(x) \quad \text{subject to:} \qquad (6)$$
$$x \in \mathscr{X}_{\delta^{(i)}}, \quad i \in \{1, 2, \ldots, N\}.$$

Since (6) has a finite number of constraints, it can be solved at low computational cost. On the other hand, the obvious question to ask is whether (6) gives a chance-constrained solution. An answer is found in the following fundamental theorem that has been established in [12].[2]

Definition 1 (violation probability). The *violation probability* of a given $x \in \mathscr{X}$ is defined as $V(x) = \mathbb{P}\{\delta \in \Delta : x \notin \mathscr{X}_\delta\}$.

Theorem 1 ([12]). *Let x_N^* be the solution to* (6). *It holds that*

$$\mathbb{P}^N\{V(x_N^*) > \varepsilon\} \leq \sum_{i=0}^{d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i}. \qquad (7)$$

From (7) one obtains that $\mathbb{P}^N\{V(x_N^*) \leq \varepsilon\} \geq 1 - \sum_{i=0}^{d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i}$, which shows that the cumulative probability distribution of $V(x_N^*)$ is bounded by a Beta distribution. This result, as all results in the scenario theory, is distribution-free, that is, it holds for all distributions $\mathbb{P}$. Moreover, it is not improvable since in [12] it is proven that the result is tight and holds with equality for a class of problems there named "fully-supported." By setting $\sum_{i=0}^{d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \beta$, the interpretation of Theorem 1 is that the scenario solution is, with (high) probability $1-\beta$, a feasible solution for a chance-constrained problem where one is allowed to violate an ε-fraction of the constraints.

To offer a more immediate understanding of the theorem, a pictorial representation of the result is given in Figure 1. In the figure, the N samples $\delta^{(1)}, \ldots, \delta^{(N)}$ are represented as a single multi-sample $(\delta^{(1)}, \ldots, \delta^{(N)})$ from Δ^N. In Δ^N there is a "bad set" represented in grey such that, if we extract a multi-sample in the bad set, then the theorem does not provide us with any conclusions. This, however, happens with tiny probability since β can be made very small, say 10^{-10}, without having to increase N excessively (this fact is discussed in [12] and it is also touched upon later in this chapter for the particular setup of MPC). In all other cases, the multi-sample maps into a finite convex optimization problem, the scenario problem, that we can easily solve and the corresponding solution automatically satisfies all the other unseen constraints except for a small fraction ε of them.

Scenario optimization has been introduced in [7], and has ever since attracted an increasing interest. Robustness properties have been studied in [8, 12, 20] and, under regularization and structural assumptions, further investigated in [2, 11,

[2] In [12], a mild assumption of existence and uniqueness of the solution (Assumption 1 in [12]) is made which we do not report here for conciseness of presentation. Moreover, paper [12] considers linear cost functions but the extension to generic convex functions is straightforward.

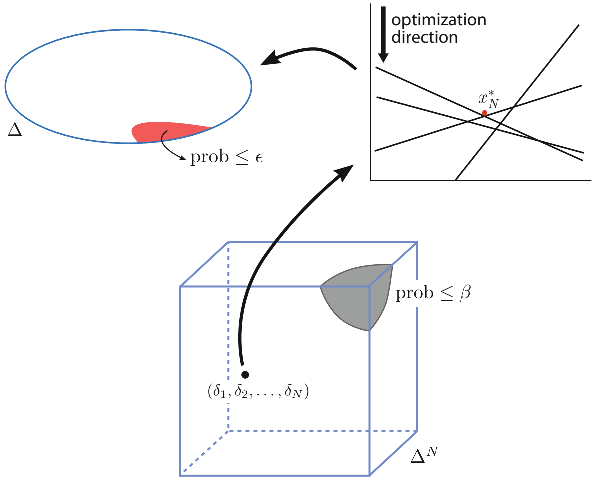

Fig. 1: Visualization of Theorem 1.

48, 55]. Papers [13, 29] consider constraints removal, and [54] examines multi-stage problems. Generalizations to a non-convex setup are proposed in [1, 28, 31]. See also [8, 16, 24, 36, 42, 51, 52] for a comparison of scenario optimization with other methods in stochastic optimization. Besides MPC, scenario optimization has found application to fields ranging from machine learning and prediction [10, 15, 22, 37] to quantitative finance [33, 39–41], from management to control design [16]. For next use in this chapter, we also recall here the main result from [29].

Theorem 2 ([29]). *Fix a value $k \le N$; remove k constraints from problem (6) according to a given, arbitrary, rule; find the solution $x^*_{k,N}$ of the so-obtained problem, and assume that the rule has been designed so that the k constraints that have been removed are violated.*[3] *It holds that*

$$\mathbb{P}^N\{V(x^*_{k,N}) > \varepsilon\} \le \binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i}. \tag{8}$$

Constraints removal is important to improve the performance of the scenario program and Theorem 2 quantifies the violation when a solution that violates k constraints is considered.

[3] Violation of the removed constraints must hold almost surely with respect to the scenario realizations and for any N, see [29] for a broad discussion.

4 The Scenario Approach for Solving Stochastic MPC

By defining the following vectors of state, input, and disturbance signals

$$\mathbf{x}_+ = \begin{bmatrix} x_{\tau+1} \\ x_{\tau+2} \\ \vdots \\ x_{\tau+M} \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} u_{\tau} \\ u_{\tau+1} \\ \vdots \\ u_{\tau+M-1} \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_{\tau} \\ w_{\tau+1} \\ \vdots \\ w_{\tau+M-1}, \end{bmatrix}$$

one can write

$$\begin{aligned} \mathbf{x}_+ &= \mathbf{F}x_\tau + \mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{w} \\ \mathbf{u} &= \Gamma + \Theta \mathbf{w}, \end{aligned} \tag{9}$$

where matrices **F**, **G**, and **H** are given by

$$\mathbf{F} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^M \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} B & 0_{n\times m} & \cdots & 0_{n\times m} \\ AB & B & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{n\times m} \\ A^{M-1}B & \cdots & AB & B \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} D & 0_{n\times l} & \cdots & 0_{n\times l} \\ AD & D & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{n\times l} \\ A^{M-1}D & \cdots & AD & D \end{bmatrix},$$

and Γ and Θ contain the parameters of the control law and are given by

$$\Gamma = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{M-1} \end{bmatrix} \quad \Theta = \begin{bmatrix} 0_{m\times l} & 0_{m\times l} & \cdots & 0_{m\times l} \\ \theta_{1,0} & 0_{m\times l} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{m\times l} \\ \theta_{M-1,0} & \cdots & \theta_{M-1,M-2} & 0_{m\times l} \end{bmatrix}.$$

Let us start by considering the constraints. Since the state and input vectors $\mathbf{x}_+$ and $\mathbf{u}$ are linear functions of the design parameters Γ and Θ (equation (9)), and function $f(\mathbf{x}_+, \mathbf{u})$ in (2) is convex, then $f(\mathbf{F}x_\tau + \mathbf{G}\Gamma + (\mathbf{H} + \mathbf{G}\Theta)\mathbf{w}, \Gamma + \Theta\mathbf{w})$ is a convex function of Γ and Θ.

As for the control cost (1), letting

$$\mathbf{Q} = \begin{bmatrix} Q_1 & \cdots & 0_{n\times n} \\ \vdots & \ddots & \vdots \\ 0_{n\times n} & \cdots & Q_M \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} R_0 & \cdots & 0_{m\times m} \\ \vdots & \ddots & \vdots \\ 0_{m\times m} & \cdots & R_{M-1} \end{bmatrix}$$

the cost can be expressed as follows:

$$\begin{aligned} J(\Gamma,\Theta) &= \mathbb{E}\left[\mathbf{x}_+^T \mathbf{Q}\mathbf{x}_+ + \mathbf{u}^T\mathbf{R}\mathbf{u}\right] \\ &= (\mathbf{F}x_\tau + \mathbf{G}\Gamma)^T\mathbf{Q}(\mathbf{F}x_\tau + \mathbf{G}\Gamma) + 2(\mathbf{F}x_\tau + \mathbf{G}\Gamma)^T\mathbf{Q}(\mathbf{H} + \mathbf{G}\Theta)\cdot\mathbb{E}[\mathbf{w}] \\ &\quad + \mathrm{tr}\left[(\mathbf{H}+\mathbf{G}\Theta)^T\mathbf{Q}(\mathbf{H}+\mathbf{G}\Theta)\cdot\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right]\right] + \Gamma^T\mathbf{R}\Gamma + 2\Gamma^T\mathbf{R}\Theta\cdot\mathbb{E}[\mathbf{w}] \\ &\quad + \mathrm{tr}\left[\Theta^T\mathbf{R}\Theta\cdot\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right]\right], \end{aligned}$$

which is a quadratic convex function of Γ and Θ.

Hence, the optimization problem to be solved at time τ can be written as the following chance-constrained optimization problem

$$\begin{aligned}&\min_{\Gamma,\Theta} J(\Gamma,\Theta) \quad \text{subject to:}\\ &\mathbb{P}\{f(\mathbf{F}x_\tau+\mathbf{G}\Gamma+(\mathbf{H}+\mathbf{G}\Theta)\mathbf{w},\Gamma+\Theta\mathbf{w})\leq 0\}\geq 1-\varepsilon.\end{aligned} \tag{10}$$

As it has been remarked in previous sections, the probabilistic constraint in (10) poses severe difficulties that can even lead to a conundrum in solving the problem. In the scenario approach, this difficulty is addressed by replacing the infinite amount of noise realizations with finitely many realizations sampled according to the noise distribution, as described in the following. Let $\mathbf{w}^{(i)}$, $i=1,2,\dots,N$, be realizations of the noise vector $\mathbf{w}$ obtained by simulating the model of the noise.[4] In this context, the scenario problem is written as

$$\begin{aligned}&\min_{\Gamma,\Theta} J(\Gamma,\Theta) \quad \text{subject to:}\\ &f(\mathbf{F}x_\tau+\mathbf{G}\Gamma+(\mathbf{H}+\mathbf{G}\Theta)\mathbf{w}^{(i)},\Gamma+\Theta\mathbf{w}^{(i)})\leq 0, \quad i\in\{1,2,\dots,N\},\end{aligned} \tag{11}$$

which corresponds to replacing the probabilistic constraint in (10) with N deterministic constraints, one for each noise realization.

Problem (11) is a standard convex optimization problem, with a convex cost $J(\Gamma,\Theta)$ and a finite number of convex constraints. Problems of this type can be efficiently solved via standard numerical solvers like those implemented in the interfaces CVX [32] or YALMIP [35]. Moreover, by using the theory presented in the previous section, one can show that the following result holds.

Theorem 3. *Select a "confidence parameter" $\beta\in(0,1)$. Then, the solution (Γ_N^*,Θ_N^*) to the scenario problem (11) satisfies the relation*

$$\mathbb{P}\{f(\mathbf{F}x_\tau+\mathbf{G}\Gamma_N^*+(\mathbf{H}+\mathbf{G}\Theta_N^*)\mathbf{w},\Gamma_N^*+\Theta_N^*\mathbf{w})\leq 0\}\geq 1-\varepsilon,$$

with probability no smaller than $1-\beta$, where (d is the number of optimization variables)

$$\varepsilon=\min\left\{\frac{2}{N}\left(\ln\frac{1}{\beta}+d\right),1\right\}. \tag{12}$$

Theorem 3 states that the scenario solution is feasible for problem (10) where ε is given by the right-hand side of (12) with confidence $1-\beta$. The scenario solution cannot be guaranteed to be always feasible because of the stochastic nature of its construction. However, infeasibility is such a rare event that it can be neglected in practice. To see this, fix ε and make N explicit in (12) with respect to ε and β, so obtaining

[4] In a standard LQG setting, this would require generating M independent Gaussian noise terms for each realization. In the scenario approach, however, there is no limitation on the noise structure and the noise can, e.g., be generated by an ARMA (Auto-Regressive Moving-Average system) or by any other model.

$$N = \frac{2}{\varepsilon}\left(\ln\frac{1}{\beta} + d\right).$$

Here, N increases logarithmically with $1/\beta$, so that enforcing a very small value of β, like $\beta = 10^{-7}$ or even $\beta = 10^{-10}$, can be done without rising N to too high values.

Formula (12) provides an explicit expression for ε as a function of N and β and can be derived from Theorem 1. Precisely, equation (12) is obtained by making explicit the right-hand side of equation (7) with respect to ε; see [2] for technical details.

Further, the cost function in the scenario problem (11) can be improved by removing some of the scenario constraints. The price to pay for this is an increase in the violation probability ε. To be precise, suppose that, from the N disturbance realizations, k realizations are removed according to Algorithm 1.[5]

The so-obtained solution satisfies the remaining $N-k$ constraints and is feasible, with probability no smaller than $1-\beta$, for the chance-constrained problem (10) with a violation probability ε as given in the next theorem, which directly follows from the right-hand side of equation (8) by making it explicit with respect to ε.

Theorem 4. *Select a 'confidence parameter' $\beta \in (0,1)$. Then, the solution $(\Gamma^*_{k,N}, \Theta^*_{k,N})$ obtained by removing k of the N constraints in (11) via Algorithm 1 satisfies the relation*

$$\mathbb{P}\{f(\mathbf{F}x_\tau + \mathbf{G}\Gamma^*_{k,N} + (\mathbf{H} + \mathbf{G}\Theta^*_{k,N})\mathbf{w}, \Gamma^*_{k,N} + \Theta^*_{k,N}\mathbf{w}) \leq 0\} \geq 1-\varepsilon,$$

with probability no smaller than $1-\beta$, where (d is the number of optimization variables)

$$\varepsilon = \min\left\{\frac{k}{N} + \frac{d+h+\sqrt{h^2+2(d+k)h}}{N}, 1\right\}, \tag{13}$$

with $h = \ln\frac{1}{\beta} + d\left(1 + \ln\frac{d+k}{d}\right)$.

In equation (13), k/N is the empirical violation probability and the guaranteed violation ε is obtained by adding a margin to it. Letting k be proportional to N, $k = \gamma N$, one obtains that the margin is $\mathrm{O}\left(\log N/\sqrt{N}\right)$, so that ε approaches $\gamma = k/N$ as N grows to infinity.

[5] Algorithm 1 is a greedy removal algorithm which is here introduced because it can be implemented at relatively low computational cost. Other alternatives exist, and the paper [13] offers an ample discussion on this matter. Algorithm 1 comes to termination provided that at each step an active constraint can be found whose elimination leads to a cost improvement. This is a very mild condition.

Algorithm 1: Scenario Algorithm with constraints removal

1: Solve problem (11) and store the solution.

2: Let i run over $1,2,\ldots,N$ and find the constraints violated by the stored solution (the first time that this point 2 is entered, the set of violated constraints is empty), that is, find the indexes i such that

$$f(\mathbf{F}x_\tau + \mathbf{G}\Gamma + (\mathbf{H}+\mathbf{G}\Theta)\mathbf{w}^{(i)}, \Gamma + \Theta\mathbf{w}^{(i)}) > 0.$$

Let these indexes be $j_1, j_2, \ldots, j_L$. **If** L is equal to k, then **halt** the algorithm and return the stored solution.

3: Find the active constraints for the stored solution, i.e., the indexes i such that

$$f(\mathbf{F}x_\tau + \mathbf{G}\Gamma + (\mathbf{H}+\mathbf{G}\Theta)\mathbf{w}^{(i)}, \Gamma + \Theta\mathbf{w}^{(i)}) = 0.$$

Let these indexes be $i_1, i_2, \ldots, i_q$.

4: **For** $h = 1,2,\ldots,q$

Solve problem

$$\min_{\Gamma,\Theta} J(\Gamma,\Theta) \quad \text{subject to:}$$

$$f(\mathbf{F}x_\tau + \mathbf{G}\Gamma + (\mathbf{H}+\mathbf{G}\Theta)\mathbf{w}^{(i)}, \Gamma + \Theta\mathbf{w}^{(i)}) \leq 0,\ i \in \{1,2,\ldots,N\}/\{i_h, j_1, j_2, \ldots, j_L\}.$$

If the obtained cost is better than the cost of the stored solution, then delete the currently stored solution and store the last computed solution.

End For

5: **Goto 2**

As said, (13) is obtained by making equation (8) explicit with respect to ε. By instead making this same equation explicit with respect to N, one sees that the smallest N so that (8) holds scales as

$$N = \mathrm{O}\left(\frac{d + \ln\frac{1}{\beta}}{(\varepsilon-\varepsilon')^2}\right),$$

where we have put $\varepsilon' = k/N$. This relation reveals some interesting features of the computational complexity of the scenario optimization algorithm. If ε' is selected to be close to the desired violation probability ε, then N becomes large. Provably, this leads to solutions that better approximate the solution to the chance-constrained problem (10); however, this is obtained at the price of an increase of the computational burden. In a given application, the choice of a suitable ε' comes from a compromise between quality of the solution and computational tractability. In many cases the extreme choice of taking $\varepsilon' = 0$ (i.e., $k = 0$, no constraint removal) already gives acceptable results.

In closing this section, one additional word deserves to be spent on the control parametrization (4). In (4), one has $d = mM + ml\frac{(M-1)M}{2}$, where mM is the number of optimization variables in Γ and $ml\frac{(M-1)M}{2}$ is the number of those in Θ. In various applications, the quadratic dependence on the horizon length M poses a hurdle in the applicability of the scenario approach due to the linear dependence of N on d. This may suggest alternative parameterizations that keep the total number of parameters lower, and some choices are illustrated below.

1. $u_{\tau+i} = \gamma_i + \sum_{j=i-r}^{i-1} \theta_{i,j} w_{\tau+j}$, which corresponds to (blank entries are zero values):

$$\Theta = \begin{bmatrix} \theta_{1,0} & & & & \\ \vdots & \ddots & & & \\ \theta_{r,0} & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \theta_{M-1,M-1-r} & \cdots & \theta_{M-1,M-2} \end{bmatrix}.$$

   In this case, $d = mM + ml\left(r(M-1-r) + \frac{(r-1)r}{2}\right)$;

2. $u_{\tau+i} = \gamma_i + \sum_{j=0}^{i-1} \theta_{i-j} w_{\tau+j}$, which corresponds to:

$$\Theta = \begin{bmatrix} \theta_1 & & & \\ \theta_2 & \ddots & & \\ \vdots & \ddots & \ddots & \\ \theta_{M-1} & \cdots & \theta_2 & \theta_1 \end{bmatrix}.$$

   In this case, $d = mM + ml(M-1)$;

3. $u_{\tau+i} = \gamma_i + \sum_{j=i-r}^{i-1} \theta_{i-j} w_{\tau+j}$, which corresponds to:

$$\Theta = \begin{bmatrix} \theta_1 & & & & \\ \vdots & \ddots & & & \\ \theta_r & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \theta_r & \cdots & \theta_1 \end{bmatrix}.$$

   In this case, $d = mM + mlr$;

4. $u_{\tau+i} = \gamma_i$, i.e., $\Theta = 0$. In this case, $d = mM$. At times, this parametrization has been combined with a fixed linear state-feedback controller,

$$u_{\tau+i} = \gamma_i + \bar{K} x_{\tau+i}, \quad \bar{K} \text{ fixed},$$

   to improve performance [18, 19].

5 Numerical Example

We consider a numerical example inspired by [21].

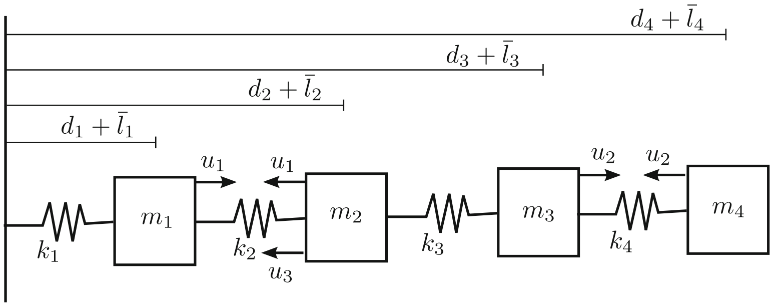

Fig. 2: Scheme of the mechanical system ($\bar{l}_1, \bar{l}_2, \bar{l}_3, \bar{l}_4$ are masses nominal positions).

The mechanical system in Figure 2 is composed by four masses and four springs. The state of the system is formed by the mass displacements, d_1, d_2, d_3, and d_4, from the nominal positions (i.e., the positions at the equilibrium when the input is zero, $\bar{l}_1$, $\bar{l}_2$, $\bar{l}_3$ and $\bar{l}_4$), and by the displacements derivatives, $\dot{d}_1$, $\dot{d}_2$, $\dot{d}_3$, and $\dot{d}_4$ (superscript dot denotes derivative). The control input is $u = [u_1, u_2, u_3]^T$, where u_1, u_2, and u_3 are forces acting on the masses as shown in Figure 2.

All masses and stiffness constants are equal to 1, i.e., $m_1 = m_2 = m_3 = m_4 = 1$ and $k_1 = k_2 = k_3 = k_4 = 1$. Assuming that the control action is held constant over the sampling period, the discrete-time model of the system is given by

$$x_{t+1} = Ax_t + Bu_t + Dw_t,$$

where

$$A = \begin{bmatrix} 0.19 & 0.35 & 0.03 & 0.00 & 0.71 & 0.14 & 0.01 & 0.00 \\ 0.35 & 0.22 & 0.35 & 0.04 & 0.14 & 0.71 & 0.14 & 0.01 \\ 0.03 & 0.35 & 0.23 & 0.39 & 0.01 & 0.14 & 0.71 & 0.14 \\ 0.00 & 0.04 & 0.39 & 0.58 & 0.00 & 0.01 & 0.14 & 0.85 \\ -1.28 & 0.44 & 0.12 & 0.01 & 0.19 & 0.35 & 0.03 & 0.00 \\ 0.44 & -1.15 & 0.45 & 0.13 & 0.35 & 0.22 & 0.35 & 0.04 \\ 0.12 & 0.45 & -1.15 & 0.57 & 0.03 & 0.35 & 0.23 & 0.39 \\ 0.01 & 0.13 & 0.57 & -0.71 & 0.00 & 0.04 & 0.39 & 0.58 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.39 & 0.00 & -0.04 \\ -0.39 & 0.04 & -0.42 \\ -0.04 & 0.39 & -0.04 \\ -0.00 & -0.42 & -0.00 \\ 0.57 & 0.01 & -0.14 \\ -0.58 & 0.13 & -0.71 \\ -0.13 & 0.57 & -0.14 \\ -0.01 & -0.71 & -0.01 \end{bmatrix},$$

and w_t is an additional stochastic disturbance that affects the system. For simplicity, in this simulation section we assume that w_t is a bi-variate white Gaussian noise with zero mean and covariance matrix $I_{2\times 2}$, and that

$$D = \begin{bmatrix} 0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix}^T,$$

which means that the external disturbance affects the fourth mass only.

The system is at rest at the initial time, that is, $x_\tau = 0$. The goal is to design a control action over a time-horizon $M = 5$ that gets the masses to be close to the nominal positions at the final time instant despite the presence of the noise. During operation, the springs are required to stay in their linear operation domain, a requirement which can be explicitly accounted for by imposing a constraint on the spring deformations, while the control action has also to satisfy saturation limits.

To be specific, we consider the average control cost (1) and set

$$Q_i = \begin{cases} 0_{8\times 8} & i < 5 \\ \left[\begin{array}{c|c} I_{4\times 4} & 0_{4\times 4} \\ \hline 0_{4\times 4} & 0_{4\times 4} \end{array}\right] & i = 5 \end{cases}, \quad \text{and } R_i = 10^{-6} I_{3\times 3}\ \forall i. \tag{14}$$

Moreover, in (3) we let $S = I$ and

$$C = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & \\ -1 & 1 & 0 & 0 & 0_{4\times 4} \\ 0 & -1 & 1 & 0 & \\ 0 & 0 & -1 & 1 & \end{array}\right],$$

so that

$$Cx_{\tau+i} = \begin{bmatrix} d_{1,\tau+i} \\ d_{2,\tau+i} - d_{1,\tau+i} \\ d_{3,\tau+i} - d_{2,\tau+i} \\ d_{4,\tau+i} - d_{3,\tau+i} \end{bmatrix}$$

represents the springs deformation at time $\tau + i$, and consider the probabilistic constraint

$$\mathbb{P}\left\{ \sup_{i=0,\dots,4} \|u_{\tau+i}\|_\infty \le 1.8 \text{ and } \sup_{i=1,\dots,5} \|Cx_{\tau+i}\|_\infty \le 1.8 \right\} \ge 1 - \varepsilon, \tag{15}$$

In this problem, using a probabilistic constraint finds justification because an excess of deformation can be tolerated as long as it does not happen too often, while relaxing the input saturation constraint leads to a less conservative design while hitting the saturation limits may rarely generate some deviation from the designed behavior.

For the actual implementation, the scenario approach was used. The control action was parameterized according to (4) (full parametrization), resulting in $d = 75$. $N = 1500$ realizations of the disturbance were sampled and the scenario problem in (11), with no removed constraints, was solved, which gave the solution (Γ_N^*, Θ_N^*). For the sake of comparison, the LQG solution, $(\Gamma_{LQG}^*, \Theta_{LQG}^*)$, was also computed by minimizing the cost with no input and state constraints. All numerical results were obtained by means of CVX, [32] with the solver MOSEK, [38].

The two cost values were $J(\Gamma_N^*, \Theta_N^*) = 0.709$ and $J(\Gamma_{LQG}^*, \Theta_{LQG}^*) = 0.481$, which gives a 31% improvement for the LQG cost. On the other hand, as expected, the LQG solution often violates constraints, and a Monte-Carlo simulation showed a 23% probability of constraints violation. In the scenario design, we have instead that, with high confidence $1 - 10^{-6}$, it holds that

$$\mathbb{P}\left\{ \sup_{i=0,\dots,4} \|u_{\tau+i}\|_\infty \le 1.8 \ \text{ and } \sup_{i=1,\dots,5} \|Cx_{\tau+i}\|_\infty \le 1.8 \right\} \ge 0.92^{6};$$

The actual probability of constraints satisfaction, computed by means of a Monte-Carlo simulation, was found to be 97%.[7]

To better appreciate the difference between the two designs (scenario and LQG), Figure 3 displays the cumulative probability distributions of $\sup_{i=0,\dots,4} \|u_{\tau+i}\|_\infty$ and of $\sup_{i=1,\dots,5} \|Cx_{\tau+i}\|_\infty$ obtained via Monte-Carlo methods when the input and state are generated by the scenario and the LQG designs.

When N is large, resorting to the scenario approach with no constraints removal returns solutions that carry high guarantees of constraints satisfaction which, however, are also poorly performing because the design is close to the worst-case (robust) design. Here, with $N = 1500$ we already had a significant decrease of performance as compared to LQG. To improve the scenario control performance, we next resorted to constraints removal and applied Algorithm 1 with $k \neq 0$. Table 1

[6] The value $\varepsilon = 0.08$ was computed from (7) by bisection instead of using the explicit formula in Theorem 3.

[7] It is perhaps worth mentioning that it is possible to obtain better evaluations of constraints satisfaction by using the results of the recent contribution [14]. Specifically, from Theorem 2 of [14], it can be proven for the present setup that, with high confidence $1 - 10^{-6}$, it holds that

$$\mathbb{P}\left\{ \sup_{i=0,\dots,4} \|u_{\tau+i}\|_\infty \le 1.8 \ \text{ and } \sup_{i=1,\dots,5} \|Cx_{\tau+i}\|_\infty \le 1.8 \right\} \ge 1 - \varepsilon(s_N^*),$$

where $\varepsilon(\cdot)$ is a function defined over the integers given in the paper and s_N^* is the number of the so-called "support constraints" that have been found in the problem at hand. In other words, $\varepsilon(s_N^*)$ is not a-priori determined and it is a-posteriori tuned to the number of support constraints. The interested reader is referred to [14] for a more-in-depth discussion. In the present simulation, it turned out that the number of support constraints was 34, resulting in $1 - \varepsilon(34) = 0.949$.

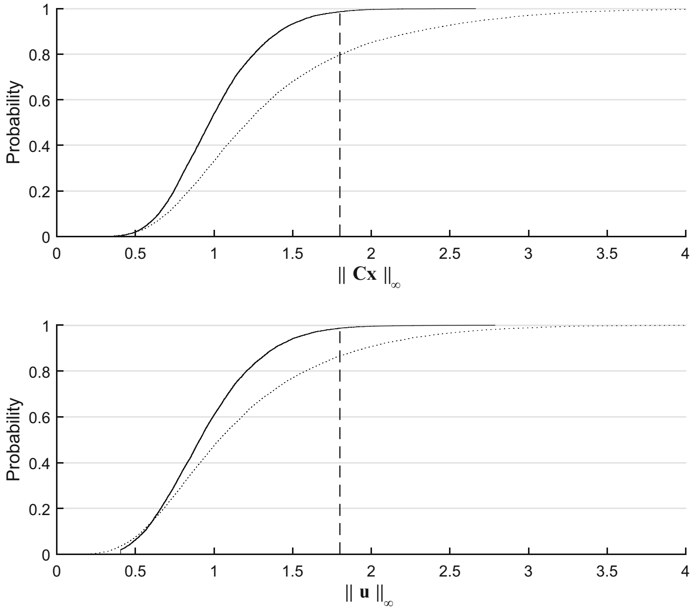

Fig. 3: Cumulative probability distributions of $\sup_{i=0,\dots,4} \|u_{\tau+i}\|_\infty$ (lower plot) and $\sup_{i=1,\dots,5} \|Cx_{\tau+i}\|_\infty$ (upper plot) for the scenario design (solid line) and LQG control (dotted line).

summarizes the results obtained for $k = 0, 10, 20, \dots, 50$. In the table, "Guaranteed prob." refers to the bound on the probability of constraints satisfaction guaranteed with confidence $1 - 10^{-6}$ and obtained from (8) by means of bisection, and "Actual prob." is the actual probability computed via Monte-Carlo methods. As it appears,

Table 1: A comparison between scenario designs with different number of removed scenarios.

k	$J(\Gamma^*_{k,N}, \Theta^*_{k,N})$	Guaranteed prob.	Actual prob.
0	0.709	0.92	0.971
10	0.661	0.881	0.967
20	0.629	0.871	0.959
30	0.597	0.863	0.952
40	0.575	0.854	0.947
50	0.564	0.845	0.937

constraint removal leads to a rapid improvement of the performance, while the probability of constraints satisfaction decreases more gently. This shows the ability of Algorithm 1 to remove portions of the uncertainty domain that have a strong impact on the cost function, a feature which is missing on LQG. Figure 4 shows the cumulative probability distributions of $\sup_{i=0,\ldots,4} \|u_{\tau+i}\|_\infty$ and of $\sup_{i=1,\ldots,5} \|Cx_{\tau+i}\|_\infty$ for $(\Gamma^*_{N,0}, \Theta^*_{N,0})$, $(\Gamma^*_{N,50}, \Theta^*_{N,50})$ and the LQG control.

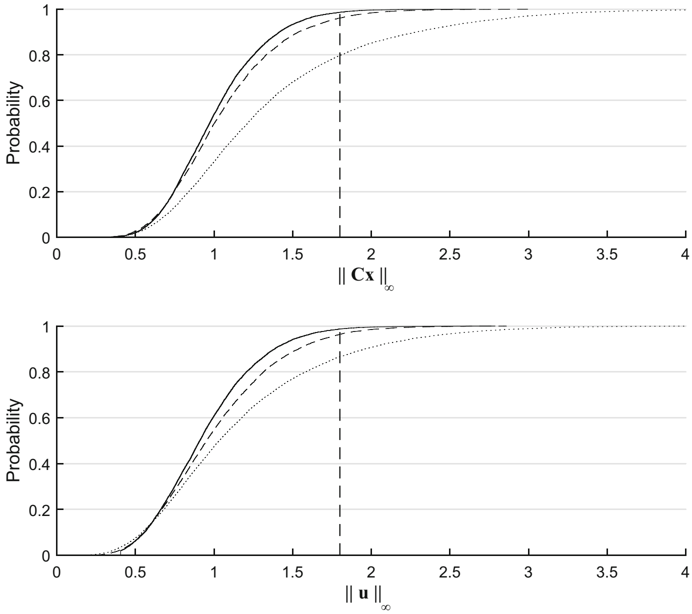

Fig. 4: Cumulative probability distributions of $\sup_{i=0,\ldots,4} \|u_{\tau+i}\|_\infty$ (lower plot) and $\sup_{i=1,\ldots,5} \|Cx_{\tau+i}\|_\infty$ (upper plot) for $(\Gamma^*_{0,N}, \Theta^*_{0,N})$ (solid line), $(\Gamma^*_{50,N}, \Theta^*_{50,N})$ (dashed line), and LQG control (dotted line).

6 Extensions and Future Work

In the present chapter, the main focus has been on the application of the scenario approach to the solution of the finite-horizon chance-constrained optimization problem (10). The resulting MPC scheme consists in the implementation of scenario optimization over a receding horizon; that is, at every time τ only the first control action u_τ is applied and, after that the system has moved to the new state $x_{\tau+1}$, the whole optimization process is repeated. This poses additional challenges that have been partly addressed in the literature and that are hinted at here.

A first issue concerns with the recursive feasibility of (11). It may happen that the stochastic noise pushes the state to a far distant condition such that a bounded input cannot succeed in satisfying the constraints in the next time period. This issue has been addressed in [25–27] by introducing a suitable relaxation to the scenario framework as described in this chapter.

A second issue refers to studying the constraint satisfaction in the long run. Often the constraint $f(x_{\tau+1},\ldots,x_{\tau+M},u_\tau,\ldots,u_{\tau+M-1}) \leq 0$ comes in the form of input or state saturation limits that apply at every point in time. For this case, the paper [49] presents a study which quantifies the asymptotic proportion of times when these limits are violated over the total number of time instants passed.

A final point that deserves to be mentioned is the possibility of applying the scenario-based MPC scheme to nonlinear systems. Nonlinearity introduces an additional difficulty relating to the convexity assumption made in this chapter since convexity fails to be true for nonlinear systems even when the function f setting the constraints is convex in its arguments. More specifically, Theorems 1 and 2 of Section 3 are grounded on the fact that, in a convex setup, the solution to (6) is determined by a limited and known number of constraints (those that are called support constraints in the literature, see [7], which are no more than the number of optimization variables). In contrast, in a non-convex setup the number of support constraints cannot be a-priori bounded and it can actually be arbitrarily large. Even though the corresponding analysis has not been fully developed at the time this chapter is being written, we envisage that the wait-and-judge perspective of [14, 17] can be used in this context to circumvent this difficulty: the support constraints are determined after the solution has been found and the evaluation of constraint violation is adapted to the found number of support constraints based on the theory of [14, 17].

References

1. Alamo, T., Tempo, R., Camacho, E.F.: A randomized strategy for probabilistic solutions of uncertain feasibility and optimization problems. IEEE Trans. Autom. Control **54**(11), 2545–2559 (2009)
2. Alamo, T., Tempo, R., Luque, A., Ramirez, D.R.: Randomized methods for design of uncertain systems: sample complexity and sequential algorithms. Automatica **51**, 160–172 (2015)
3. Batina, I. Model predictive control for stochastic systems by randomized algorithms. PhD thesis, Technische Universiteit Eindhoven (2004)
4. Bemporad, A., Morari, M.: Robust model predictive control: a survey. In: Robustness in Identification and Control. Lecture Notes in Control and Information Sciences, vol. 245, pp. 207–226. Springer, Berlin (1999)
5. Bemporad, A., Borrelli, F., Morari, M.: Min-max control of constrained uncertain discrete-time linear systems. IEEE Trans. Autom. Control **48**(9), 1600–1606 (2003)
6. Ben-Tal, A., Boyd, S., Nemirovski, A.: Extending scope of robust optimization: comprehensive robust counterparts of uncertain problems. Math. Program. **107**(1–2), 63–89 (2006)
7. Calafiore, G., Campi, M.C.: Uncertain convex programs: randomized solutions and confidence levels. Math. Program. **102**(1), 25–46 (2005)
8. Calafiore, G., Campi, M.C.: The scenario approach to robust control design. IEEE Trans. Autom. Control **51**(5), 742–753 (2006)
9. Calafiore, G.C., Fagiano, L.: Robust model predictive control via scenario optimization. IEEE Trans. Autom. Control **58**(1), 219–224 (2013)
10. Campi, M.C.: Classification with guaranteed probability of error. Mach. Learn. **80**, 63–84 (2010)
11. Campi, M.C., Carè, A.: Random convex programs with L1-regularization: sparsity and generalization. SIAM J. Control Optim. **51**(5), 3532–3557 (2013)

12. Campi, M.C., Garatti, S.: The exact feasibility of randomized solutions of uncertain convex programs. SIAM J. Optim. **19**(3), 1211–1230 (2008)
13. Campi, M.C., Garatti, S.: A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. J. Optim. Theory Appl. **148**(2), 257–280 (2011)
14. Campi, M.C., Garatti, S.: Wait-and-judge scenario optimization. Math. Program. **167**(1), 155–189 (2018)
15. Campi, M.C., Calafiore, G., Garatti, S.: Interval predictor models: identification and reliability. Automatica **45**(2), 382–392 (2009)
16. Campi, M.C., Garatti, S., Prandini, M.: The scenario approach for systems and control design. Annu. Rev. Control **33**(2), 149–157 (2009)
17. Campi, M.C., Garatti, S., Ramponi, F.: Non-convex scenario optimization with application to system identification. In: Proceedings of the 54th IEEE Conference on Decision and Control, Osaka (2015)
18. Cannon, M., Kouvaritakis, B., Wu, X.: Model predictive control for systems with stochastic multiplicative uncertainty and probabilistic constraints. Automatica **45**, 167–172 (2009)
19. Cannon, M., Kouvaritakis, B., Wu, X.: Probabilistic constrained MPC for multiplicative and additive stochastic uncertainty. IEEE Trans. Autom. Control **54**(7), 1626–1632 (2009)
20. Carè, A., Garatti, S., Campi, M.C.: Scenario min-max optimization and the risk of empirical costs. SIAM J. Optim. **25**(4), 2061–2080 (2015)
21. Cinquemani, E., Agarwal, M., Chatterjee, D., Lygeros, J.: Convexity and convex approximations of discrete-time stochastic control problems with constraints. Automatica **47**(9), 2082–2087 (2011)
22. Crespo, L.G., Giesy, D.P., Kenny, S.P.: Interval predictor models with a formal characterization of uncertainty and reliability. In: Proceedings of the 53rd IEEE Conference on Decision and Control (CDC), Los Angeles, pp. 5991–5996 (2014)
23. de la Peña, D.M., Alamo, T., Bemporad, A., Camacho, F.: A decomposition algorithm for feedback min-max model predictive control. IEEE Trans. Autom. Control **51**(10), 1688–1692 (2006)
24. de Mello, T.H., Bayraksan, G.: Monte Carlo sampling-based methods for stochastic optimization. Surv. Oper. Res. Manag. Sci. **19**(1), 56–85 (2014)
25. Deori, L., Garatti, S., Prandini, M.: Stochastic constrained control: trading performance for state constraint feasibility. In: Proceedings of the 12th European Control Conference, Zurich (2013)
26. Deori, L., Garatti, S., Prandini, M.: Stochastic control with input and state constraints: a relaxation technique to ensure feasibility. In: Proceedings of the 54th IEEE Conference on Decision and Control, Osaka (2015)
27. Deori, L., Garatti, S., Prandini, M.: Trading performance for state constraint feasibility in stochastic constrained control: a randomized approach. J. Frankl. Inst. **354**(1), 501–529 (2017)
28. Esfahani, P.M., Sutter, T., Lygeros, J.: Performance bounds for the scenario approach and an extension to a class of non-convex programs. IEEE Trans. Autom. Control **60**(1), 46–58 (2015)
29. Garatti, S., Campi, M.C.: Modulating robustness in control design: principles and algorithms. IEEE Control Syst. Mag. **33**(2), 36–51 (2013)
30. Goulart, P.J., Kerrigan, E.C., Maciejowski, J.M.: Optimization over state feedback policies for robust control with constraints. Automatica **42**(4), 523–533 (2006)
31. Grammatico, S., Zhang, X., Margellos, K., Goulart, P.J., Lygeros, J.: A scenario approach for non-convex control design. IEEE Trans. Autom. Control **61**(2), 334–345 (2016)
32. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, Feb 2011
33. Hong, L.J., Hu, Z., Liu, G.: Monte Carlo methods for value-at-risk and conditional value-at-risk: a review. ACM Trans. Model. Comput. Simul. **24**(4), 22:1–22:37 (2014)
34. Kothare, M., Balakrishnan, V., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica **32**(10), 1361–1379 (1996)

35. Löfberg, J.: YALMIP: a toolbox for modeling and optimization in MATLAB. In: Proceedings of the CACSD Conference, Taipei (2004)
36. Margellos, K., Goulart, P.J., Lygeros, J.: On the road between robust optimization and the scenario approach for chance constrained optimization problems. IEEE Trans. Autom. Control **59**(8), 2258–2263 (2014)
37. Margellos, K., Prandini, M., Lygeros, J.: On the connection between compression learning and scenario based single-stage and cascading optimization problems. IEEE Trans. Autom. Control **60**(10), 2716–2721 (2015)
38. MOSEK ApS: The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28) (2015)
39. Pagnoncelli, B.K., Vanduffel, S.: A provisioning problem with stochastic payments. Eur. J. Oper. Res. **221**(2), 445–453 (2012)
40. Pagnoncelli, B.K., Ahmed, S., Shapiro, A.: Sample average approximation method for chance constrained programming: theory and applications. J. Optim. Theory Appl. **142**(2), 399–416 (2009)
41. Pagnoncelli, B.K., Reich, D., Campi, M.C.: Risk-return trade-off with the scenario approach in practice: a case study in portfolio selection. J. Optim. Theory Appl. **155**(2), 707–722 (2012)
42. Petersen, I.R., Tempo, R.: Robust control of uncertain systems: classical results and recent developments. Automatica **50**, 1315–1335 (2014)
43. Prandini, M., Garatti, S., Lygeros, J.: A randomized approach to stochastic model predictive control. In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), December 2012, pp. 7315–7320
44. Prékopa, A.: Stochastic Programming. Kluwer, Boston (1995)
45. Prèkopa, A.: Probabilistic programming. In: Ruszczyǹski, A., Shapiro, A. (eds.) Stochastic Programming, volume 10 of Handbooks in Operations Research and Management Science, pp. 267–352. Elsevier, London (2003)
46. Primbs, J.A., Sung, C.H.: Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. IEEE Trans. Autom. Control **54**(2), 221–230 (2009)
47. Raimondo, D.M., Limon, D., Lazar, M., Magni, L., Camacho, E.F.: Min-max model predictive control of nonlinear systems: a unifying overview on stability. Eur. J. Control **15**, 5–21 (2009)
48. Schildbach, G., Fagiano, L., Morari, M.: Randomized solutions to convex programs with multiple chance constraints. SIAM J. Optim. **23**(4), 2479–2501 (2013)
49. Schildbach, G., Fagiano, L., Frei, C., Morari, M.: The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. Automatica **50**(12), 3009–3018 (2014)
50. Scokaert, P.O.M., Mayne, D.Q.: Min-max feedback model predictive control for constrained linear systems. IEEE Trans. Autom. Control **43**, 1136–1142 (1998)
51. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on stochastic programming: modeling and theory. MPS-SIAM, Philadelphia (2009)
52. Tempo, R., Calafiore, G., Dabbene, F.: Randomized Algorithms for Analysis and Control of Uncertain Systems, 2nd edn. Springer, London (2013)
53. van Hessem, D.H., Bosgra, O.H.: Stochastic closed-loop model predictive control of continuous nonlinear chemical processes. J. Proc. Control **16**(3), 225–241 (2006)
54. Vayanos, P., Kuhn, D., Rustem, B.: A constraint sampling approach for multistage robust optimization. Automatica **48**(3), 459–471 (2012)
55. Zhang, X., Grammatico, S., Schildbach, G., Goulart, P.J., Lygeros, J.: On the sample size of random convex programs with structured dependence on the uncertainty. Automatica **60**, 182–188 (2015)