POLITECNICO DI TORINO
Repository ISTITUZIONALE

Computationally efficient stochastic MPC: A probabilistic scaling approach

Original
Computationally efficient stochastic MPC: A probabilistic scaling approach / Mammarella, M.; Alamo, T.; Dabbene, F.; Lorenzen, M.. - ELETTRONICO. - (2020), pp. 25-30. (Intervento presentato al convegno 4th IEEE Conference on Control Technology and Applications, CCTA 2020 tenutosi a Montreal, Canada nel 2020) [10.1109/CCTA41146.2020.9206383].

Availability:
This version is available at: 11583/2907169 since: 2021-06-16T11:20:44Z

Publisher:
IEEE

Published
DOI:10.1109/CCTA41146.2020.9206383

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

19 April 2024

Computationally efficient stochastic MPC: a probabilistic scaling approach

Martina Mammarella[1], Teodoro Alamo[2], Fabrizio Dabbene[1,*], and Matthias Lorenzen[3]

Abstract— **In recent years, the increasing interest in stochastic model predictive control (SMPC) schemes has highlighted the limitation arising from their inherent computational demand, which has restricted their applicability to slow-dynamics and high-performing systems. To reduce the computational burden, in this paper we extend the probabilistic scaling approach to obtain a low-complexity inner approximation of chance-constrained sets. This approach provides probabilistic guarantees at a lower computational cost than other schemes for which the sample complexity depends on the design space dimension. To design candidate simple approximating sets, which approximate the shape of the probabilistic set, we introduce two possibilities: i) fixed-complexity polytopes, and ii) ℓ_p-norm based sets. Once the candidate approximating set is obtained, it is scaled around its center so to enforce the expected probabilistic guarantees. The resulting scaled set is then exploited to enforce constraints in the classical SMPC framework. The computational gain obtained with respect to the scenario approach is demonstrated via simulations, where the objective is the control of a fixed-wing UAV performing a crop-monitoring mission over a sloped vineyard.**

I. Introduction

In recent years, the performance degradation of model predictive control (MPC) schemes in the presence of uncertainty has driven the interest towards stochastic MPC (SMPC), to overcome the inherent conservativeness of robust approaches. A probabilistic description of the disturbance or uncertainty allows to optimize the average performance or appropriate risk measures. Furthermore, allowing a (small) probability of constraint violation, by introducing so-called chance constraints, seems more appropriate in some applications, as described in [1]. The craved control performance and constraint satisfaction can be guaranteed by properly generating a sufficient number of uncertainty realization and solving a suitable constrained optimization problem, as proposed in [2], [3]. The main advantage of this class of SMPC algorithms is given by the inherent flexibility to be applied to (almost) every class of systems, including any type of uncertainty and both state and input constraints, as long as the optimization problem is convex. On the other hand, they share the main drawback: a significant computational burden required for real-time implementation, narrowing the application domains to those involving low-computation assets and slow dynamics, with sample time measured in tens of seconds or minutes. Some examples are described in [4] for water networks, in [5] for river flood control, in [6] for chemical processes, and in [7] for energy plants.

An efficient solution to the aforementioned disadvantages was proposed in [8], where the SMPC controller design is based on an *offline* sampling approach and only a predefined number of necessary samples are kept for online implementation. In this approach, the sample complexity is linearly dependent to the design space dimension and the sampling procedure allows to obtain offline an inner approximation of the chance-constrained set. This approach has been extended to a more generic setup in [9] and experimentally validated for the control of a spacecraft during rendezvous maneuvers.

Among challenging applications, the control of unmanned aerial vehicles (UAVs) during assorted scenarios, have been triggering the attention of MPC community. These platforms are typically characterized by fast dynamics and equipped with computationally-limited autopilots. In the last decade, different receding horizon techniques have been proposed, see e.g. [10], [11], [12], [13], including a stochastic approach by [14]. In this case, preliminary analysis have confirmed the need to further reduce the dimension of the optimization problem to comply with faster dynamics and low-cost, low-performance hardware.

The main contribution of this paper is to propose a new methodology that combines the probabilistic-scaling approach proposed in [15], which allows to obtain a low-complexity inner approximation of the chance constrained set, with the SMPC approach of [8], [9]. In [15], authors show how to scale a given set of manageable complexity around its center to obtain, with a user-defined probability, a region that is included in the chance constrained set. In this paper, we extend the aforementioned approach showing how it is possible to reduce the sample complexity via probabilistic scaling exploiting so-called simple approximating sets (SAS). The starting point consists in obtaining a first simple approximation of the "shape" of the probabilistic set. To design a candidate SAS, we propose two possibilities: i) based on the definition of an approximating set by drawing a fixed number of samples and ii) envisioning ℓ_p-norm based sets, first proposed in [16]. Solving a standard optimization problem, it is possible to obtain the center and the shape of the SAS, which will later be scaled to obtain the expected probabilistic guarantees following the approach described in

This work was funded by the Italian IIT and MIUR within the 2017 PRIN (N. 2017S559BB) and by the Spanish Ministerio de Ciencia e Innovación (PID2019-106212RB-C41).

[1] Institute of Electronics, Computer and Telecommunication Engineering, National Research Council of Italy, Turin, Italy, martina.mammarella@ieiit.cnr.it, fabrizio.dabbene@ieiit.cnr.it

[2] Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n, 41092 Sevilla, Spain, talamo@us.es

[3] lorenzen@ist.uni-stuttgart.de

[*] corresponding author.

[15]. Then, the scaled SAS is used in the classical SMPC algorithm to enforce constraints.

The proposed approach is validated in an agriculture scenario, in which a fixed-wing UAV performs a crop-monitoring mission over a sloped vineyard. The example has been chosen due to the increasing interest of using drones in an agriculture 4.0 framework, as explained in [17]. The technology has a great potential to support and address some of the most pressing dares in farming. The performance of the proposed probabilistic-scaling SMPC (PS-SMPC) approach in terms of tracking capabilities and computational load has been compared with those obtained exploiting the offline sampling SMPC (OS-SMPC) proposed in [14].

Notation: The set $\mathbb{N}_{>0}$ denotes the positive integers, the set $\mathbb{N}_{\geq 0} = \{0\} \cup \mathbb{N}_{>0}$ the non-negative integers, and $\mathbb{N}_a^b$ the integers in the interval $[a, b]$. Positive (semi-) definite matrices A are denoted $A \succ 0$ ($A \succeq 0$) and $\|x\|_A^2 \doteq x^\top Ax$. $[H]_j^T$ denotes the j-th row of a matrix or vector H. Pr_a denotes the probabilistic distribution of a random variable a. Sequence of scalars/vectors are denoted with bold lower-case letters, e.g. $\mathbf{v}$.

II. Offline Sampling-based Stochastic MPC

In this section, we first recall the SMPC framework proposed in [8] and [9].

A. Problem setup

We consider the case of a discrete-time system subject to a generic uncertainty $w_k \in \mathbb{R}^{n_w}$

$$x_{k+1} = A(w_k)x_k + B(w_k)u_k + a_w(w_k), \tag{1}$$

with state $x_k \in \mathbb{R}^n$, control input $u_k \in \mathbb{R}^m$, and the vector valued function $a_w(w_k)$ representing the additive disturbance affecting the system states. The system matrices $A(w_k)$ and $B(w_k)$, of appropriate dimensions, are (possibly nonlinear) functions of the uncertainty w_k at step k. The disturbances $(w_k)_{k \in \mathbb{N}_{\geq 0}}$ are modeled as realizations of the stochastic process $(W_k)_{k \in \mathbb{N}_{\geq 0}}$, on which we take the following assumptions.

Assumption 1 (Random Disturbances): The disturbances W_k, for $k \in \mathbb{N}_{\geq 0}$, are independent and identically distributed (i.i.d.), zero-mean random variables with support $\mathbb{W} \subseteq \mathbb{R}^{n_w}$. Moreover, given $\mathbb{G} = \{(A(w_k), B(w_k), a_w(w_k))\}_{w_k \in \mathbb{W}}$, a polytopic outer approximation $\bar{\mathbb{G}} \doteq co\{A^j, B^j, a_w^j\}_{j \in \mathbb{N}_1^{N_c}} \supseteq \mathbb{G}$ with N_c vertices exists and is known.

The assumption of independent random variables is necessary to perform the offline computations discussed next. Note that the system can be augmented by a filter, to model a specific stochastic processes of interest. A known outer bound is assumed to establish a safe operating region, in which recursive feasibility of the MPC optimization can be established, see [8] for details. We remark that the system representation in (1) is very general, and encompasses e.g., those in [8], [9], [18]. Given the model (1) and a realization of the state x_k at time k, state *predictions* l steps ahead are random variables and are denoted $x_{l|k}$, to differentiate it from the *realization* x_{l+k}. Similarly, $u_{l|k}$ denotes predicted inputs that are computed based on the realization of the state x_k.

The system is subject to p state and input chance constraints of the form

$$\mathsf{Pr}_{\mathbf{w}} \left\{ [H_x]_j^T x_{l|k} + [H_u]_j^T u_{l|k} \leq 1 \mid x_k \right\} \geq 1 - \varepsilon_j, \quad l \in \mathbb{N}_{>0},\ j \in \mathbb{N}_1^p, \tag{2}$$

with $\varepsilon_j \in (0, 1)$, $H_x \in \mathbb{R}^{p \times n}$, and $H_u \in \mathbb{R}^{p \times m}$. The probability $\mathsf{Pr}_{\mathbf{w}}$ is measured with respect to the sequence $\mathbf{w} = \{w_i\}_{i \geq k}$. Hence, equation (2) states that the probability of violating the linear constraint $[H_x]_j^T x + [H_u]_j^T u \leq 1$ for any future realization of the disturbance should not be larger than ε_j. Note that *hard* input constraints can also be formulated in a similar framework, cf. [8].

The objective is to derive an asymptotically stabilizing control law for the system (1) such that, in closed loop, the constraints (2) are satisfied.

B. Stochastic Model Predictive Control

To solve the constrained control problem, an SMPC algorithm is considered. The approach is based on repeatedly solving a stochastic optimal control problem over a finite, moving horizon, but implementing only the first control action. Define the control sequence as $\mathbf{u}_k = (u_{0|k}, u_{1|k}, \ldots, u_{T-1|k})$, the prototype optimal control problem that is to be solved at each sampling time is obtained minimizing the cost function

$$J_T(x_k, \mathbf{u}_k) = \mathbb{E}\left\{ \sum_{l=0}^{T-1} \left(x_{l|k}^\top Q x_{l|k} + u_{l|k}^\top R u_{l|k} \right) + x_{T|k}^\top P x_{T|k} \mid x_k \right\} \tag{3}$$

with $Q \succ 0$, $R \succ 0$, and appropriately chosen $P \succ 0$, subject to the system dynamics (1) and constraints (2).

The online solution of the SMPC problem remains a challenging task but several special cases, which can be evaluated exactly, as well as methods to approximate the general solution have been proposed in the literature. The approach followed in this work was first proposed in [8], [9], where an offline sampling scheme was introduced. Therein, with an input parametrization

$$u_{l|k} = Kx_{l|k} + v_{l|k}, \tag{4}$$

with a prestabilizing control gain $K \in \mathbb{R}^{n \times m}$ and free optimization variables $v_{l|k} \in \mathbb{R}^m$, equation (1) is solved explicitly for the predicted states $x_{1|k}, \ldots, x_{T|k}$ and predicted inputs $u_{0|k}, \ldots, u_{T-1|k}$. In this case, the expected value of the finite-horizon cost (3) can be evaluated *offline*, leading to a quadratic cost function of the form

$$J_T(x_k, \mathbf{v}_k) = [x_k^T\ \mathbf{v}_k^T\ \mathbf{1}_n^T] \tilde{S} \begin{bmatrix} x_k \\ \mathbf{v}_k \\ \mathbf{1}_n \end{bmatrix} \tag{5}$$

in the deterministic variables $\mathbf{v}_k = (v_{0|k}, v_{1|k}, \ldots, v_{T-1|k})$ and x_k. The reader is referred to [9, Appendix A] for a detailed derivation of the cost matrix $\tilde{S}$.

Similarly, with $\mathbf{w}_k = \{w_l\}_{l=k,\ldots,k+T-1}$, the j-th chance constraint defined in (2) becomes the linear chance constraint

$$\mathbb{X}_\varepsilon^j = \left\{ \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \in \mathbb{R}^{n+mT} \mid \right. \\ \left. \mathsf{Pr}_{\mathbf{w}_k} \left\{ f_j^T(\mathbf{w}_k) \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \leq 1 \right\} \geq 1-\varepsilon \right\}, \quad (6)$$

on the new optimization variables $\mathbf{v}_k$ with f_j being a function of the sequence of random variables $\mathbf{w}_k$. Similarly to the matrix $\tilde{S}$, functions f_j are derived by explicitly solving the system dynamics over the prediction horizon and the reader is referred to [9] for a detailed derivation.

The advantage of this explicit formulation is that, given independence of the random variables, an inner approximation of the chance constraints can be derived *offline*. In [8], results from statistical learning theory (cf. [19], [20]) are exploited to construct an inner approximation $\underline{\mathbb{X}}^j$ of the constraint set $\mathbb{X}_\varepsilon^j$ by extracting N_{LT} i.i.d. samples $\mathbf{w}_k^{(i)}$ of $\mathbf{w}_k$ and taking the intersection of the sampled constraints, i.e.,

$$\underline{\mathbb{X}}_{LT}^j = \left\{ \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \in \mathbb{R}^{n+mT} \mid \right. \\ \left. f_j^T(\mathbf{w}_k^{(i)}) \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \leq 1,\ i=1,\ldots,N_{LT} \right\}. \quad (7)$$

In particular, it has been shown in [8], that for given probabilistic levels $\delta \in (0,1)$ and $\varepsilon_j \in (0, 0.14)$, choosing the sample complexity $N_{LT}^j \geq \tilde{N}(d, \varepsilon_j, \delta)$ with

$$\tilde{N}(d,\varepsilon_j,\delta) \doteq \frac{4.1}{\varepsilon_j}\Big(\ln\frac{21.64}{\delta} + 4.39d\,\log_2\Big(\frac{8e}{\varepsilon_j}\Big)\Big), \quad (8)$$

with $d = n + mT$, guarantees that with probability at least δ the sample approximation $\underline{\mathbb{X}}_{LT}^j$ is a subset of the original chance constraint $\mathbb{X}_\varepsilon^j$, i.e.,

$$\mathsf{Pr}\left\{\underline{\mathbb{X}}_{LT}^j \subseteq \mathbb{X}_\varepsilon^j\right\} \geq 1-\delta, \quad j=1,\ldots,p. \quad (9)$$

Exploiting these results, the SMPC problem can be approximated conservatively by the linearly constrained quadratic program

$$\begin{aligned} &\min_{\mathbf{v}_k}\ J_T(x_k,\mathbf{v}_k) \\ &\text{s.t. } (x_k,\mathbf{v}_k) \in \underline{\mathbb{X}}_{LT}^j, \quad j=1,\ldots,p. \end{aligned} \quad (10)$$

While the result reduces the original stochastic optimization program to an efficiently solvable quadratic program, the ensuing number of constraints, equal to

$$N_{LT} = \sum_{i=1}^{T} N_{LT}^j,$$

may still be too large. For instance, even for a moderately sized MPC problem with $n = 5$ states, $m = 2$ inputs and horizon of $T = 10$, and for a reasonable choice of probabilistic $\varepsilon_j = 0.05$, $\delta = 10^{-6}$, we get $N_{LT}^j = 20,604$. For this reason, in [8] a post-processing step for the constraint set was proposed to remove redundant constraints. While it is indeed true that all the cumbersome computations may be performed offline, it is still the case that in applications with stringent requirements on the solution time the final number of inequalities may easily become unbearable. This observation motivates the approach presented in the next section, which builds upon the results presented in [15]. We show how the probabilistic scaling approach directly leads to approximations of user chosen complexity and which can be directly used in applications instead of creating the need for a post processing step to reduce the complexity of the sampled set.

III. Complexity Reduction via Probabilistic Scaling

In this section, we focus on efficiently solving the SMPC optimization problem. To this end, we consider the more generic problem of finding a decision variable vector ξ, restricted to a set $\Xi \subseteq \mathbb{R}^{n_\xi}$, subject to p uncertain linear inequalities. Formally, we consider uncertain inequalities of the form

$$F(q)\xi \leq g(q) \quad (11)$$

where $F(q) \in \mathbb{R}^{p\times n_\xi}$ and $g(q) \in \mathbb{R}^{n_p}$ are continuous functions of the uncertainty vector $q \in \mathbb{R}^{n_q}$. The uncertainty vector q is assumed to be of random nature, with given probability distribution Pr_q and (possibly unbounded) support $\mathbb{Q}$. Hence, to each sample of q corresponds a different set of linear inequalities. We aim at finding an approximation of the ε-chance-constraint set, defined as

$$\mathbb{X}_\varepsilon \doteq \Big\{\xi \in \Xi \mid \mathsf{Pr}_q\left\{F(q)\xi \leq g(q)\right\} \geq 1-\varepsilon\Big\} \quad (12)$$

that represents the region of the design space Ξ for which this probabilistic constraint is satisfied. Note that this captures exactly the SMPC setup discussed in the previous section. Indeed, the chance-constrained set in (6) is a special instance of (12), with $\xi = [x_k^T \quad \mathbf{v}_k^T]^T$ and $q = \mathbf{w}_k$.

The characterization of the chance constrained set has several applications in robust and stochastic control. A classical approach is to find an inner convex approximation of the probabilistic set $\mathbb{X}_\varepsilon$, obtained for instance by means of applications of Chebyshev-like inequalities, see e.g. [21] and [22]. A recent approach, which is the one applied in the previous section to the SMPC problem, is instead based on the derivation of probabilistic approximations of the chance constraints set $\mathbb{X}_\varepsilon$ through sampling of the uncertainty. That is, we aim at constructing a set $\underline{\mathbb{X}}$, which is contained in $\mathbb{X}_\varepsilon$ *with high probability*.

Denote $F_j(q)$ and g_j the j-th row of $F(q)$ and j-th component of q, respectively. Consider the binary functions

$$h_j(\xi,q) \doteq \begin{cases} 0 & \text{if } F_j(q)\xi \leq g_j(q) \\ 1 & \text{otherwise} \end{cases}, \ j=1,\ldots,p.$$

Now, if we define

$$h(\xi,q) \doteq \max_{j=1,\ldots,p} h_j(\xi,q),$$

we have that h is an $(1,p)$-boolean function since it can be expressed as a function of p boolean functions, each of them involving a polynomial of degree 1. See e.g. [20, Definition 7] for a precise definition of this sort of boolean functions. Suppose that we draw N i.i.d. samples $q^{(i)}$, $i = 1,\ldots,N$. Then, we can consider the (empirical) region $\mathbb{X}_N$ defined as

$$\mathbb{X}_N \doteq \{\ \xi \in \mathbb{R}^{n_\xi}\ :\ h(\xi,q^{(i)}) = 0,\ i=1,\ldots,N\ \}.$$

It has been proved in [20, Theorem 8], that if $\epsilon \in (0, 0.14)$ and N is chosen such that[1]

$$N \geq \frac{4.1}{\epsilon}\left(\ln \frac{21.64}{\delta} + 4.39 n_\xi \log_2\left(\frac{8ep}{\epsilon}\right)\right)$$

then $\mathbb{X}_N \subseteq \mathbb{X}_\varepsilon$ with a probability no smaller than $1-\delta$.

We notice that $\mathbb{X}_N$ is a convex set, which is a desirable property in an optimization framework. However, the number of required samples N might be prohibitive for a real-time application. To tackle this issue, in this paper we exploit an appealing alternative approach proposed in [15], and we specialize it to the problem at hand. This work proposes a probabilistic scaling approach to obtain, with given confidence, an inner approximation of the chance constrained set $\mathbb{X}_\varepsilon$ avoiding the computational burden due to the sample complexity raising in other strategies.

The main idea behind this approach consist in first obtaining a simple initial approximation of the "shape" of the probabilistic set $\mathbb{X}_\varepsilon$ by exploiting simple approximating sets of the form $x_c \oplus \underline{\mathbb{S}}$. This set is not required to have *any* guarantees of probabilistic nature. Instead, to derive such probabilistic guaranteed set, a scaling procedure is devised. In particular, an optimal scaling factor γ is derived so that the set scaled around its center x_c,

$$\underline{\mathbb{S}}(\gamma) \doteq x_c \oplus \gamma \underline{\mathbb{S}}. \tag{13}$$

is guaranteed to be an inner approximation of $\mathbb{X}_\varepsilon$ with the desired confidence level δ.

A. Simple Approximating Sets

The idea at the basis of the proposed approach is to define Simple Approximating Sets (SAS), which represent specifically defined sets with a low – and pre-defined – number of constraints. First, we note that the most straightforward way to design a candidate SAS is to draw a fixed number N_S of uncertainty samples, and to construct a sampled approximation as follows:

1. Sampled-poly

$$\underline{\mathbb{S}}_S = \bigcap_{i=1}^{N_S} \mathbb{X}_i = \bigcap_{i=1}^{N_S} \left\{ \xi \in \Xi \mid F(q^{(i)})\xi \leq g(q^{(i)}),\ i = \mathbb{N}_1^{N_S} \right\} \tag{14}$$

Clearly, if $N_S << N_{LT}$, the probabilistic properties of $\underline{\mathbb{S}}_S$ before scaling will be very bad. To tackle this unwanted behavior, a probabilistic scaling approach is proposed and is detailed in Section III-B.

A second way to construct a SAS considered in this paper exploits a class of ℓ_p-norm based sets introduced in [16] as follows

$$\mathcal{A}(x_c, P) \doteq \{\xi \in \mathbb{R}^{n_\xi} \mid \xi = x_c + Pz, z \in \mathcal{B}_p\}, \tag{15}$$

where $\mathcal{B}_p \subset \mathbb{R}^{n_\xi}$ is the unit ball in the p norm, x_c is the center and $P = P^T \succeq 0$ is the so-called *shape* matrix. In particular, we note that for $p = 1, \infty$ these sets take the form of polytopes with fixed number of facets/vertices. Hence, we introduce the following two SAS:

2. ℓ_1**-poly**, defined starting from a *cross-polytope*, also known as *diamond*, of order n_ξ with $2n_\xi$ vertices and 2^{n_ξ} facets, i.e.

$$\underline{\mathbb{S}}_1 = \{\xi \in \mathbb{R}^{n_\xi} \mid \xi = x_c + Pz,\ \|z\|_1 \leq 1\}. \tag{16}$$

3. ℓ_∞**-poly**, defined starting from a *hyper-cube* of dimension n_ξ with 2^{n_ξ} vertices and $2n_\xi$ facets, i.e.

$$\underline{\mathbb{S}}_\infty = \{\xi \in \mathbb{R}^{n_\xi} \mid \xi = x_c + Pz,\ \|z\|_\infty \leq 1\}. \tag{17}$$

Hence, the problem becomes designing the center and shape parameters (x_c, P) of the set $\underline{\mathbb{S}}_1$ (resp. $\underline{\mathbb{S}}_\infty$) so that they represent in the best possible way the set $\mathbb{X}_\varepsilon$. To this end, we start from a *sampled design polytope*

$$\mathbb{D} = \bigcap_{i=1}^{N_D} \mathbb{X}_i,$$

with a fixed number of samples N_D, and construct the largest set $\underline{\mathbb{S}}_1$ (resp. $\underline{\mathbb{S}}_\infty$) contained in $\mathbb{D}$. It is easily observed that to obtain the largest ℓ_1-poly inscribed in $\mathbb{D}$, we need to solve the following convex optimization problem

$$\begin{aligned} \max_{x_c, C} \quad & \mathrm{tr}(P) \\ \text{s.t.} \quad & P \succeq 0, \\ & f_i^T P z^{[j]} \leq g_i - f_i^T x_c, \quad i = \mathbb{N}_1^{N_D}, \quad z^{[j]} \in \mathcal{V}_1, \end{aligned} \tag{18}$$

where $\mathcal{V}_1 = \{z^{[1]}, \ldots, z^{[2n_\xi]}\}$ are the vertices of the unit cross-polytope while the vertices of the optimal ℓ_1-poly can then be obtained as

$$\xi^{[j]} = x_c + Pz^{[j]}, \quad j = 1, \ldots, 2n_\xi. \tag{19}$$

It should be remarked that, from these vertices, one could then recover the corresponding 2^{n_ξ} linear inequalities, each one defining a facet of the rotated diamond. However, this procedure, besides being computationally extremely demanding (going from a vertex-description to a linear inequality description of a polytope is known to be NP hard, [23]), would lead to an exponential number of linear inequalities, thus rendering the whole approach not viable. Instead, we exploit the following equivalent formulation of (15), see e.g. [16] for details

$$\underline{\mathbb{S}}_1 = \{\xi \in \mathbb{R}^{n_\xi} \mid \|M\xi - c\|_1 \leq 1\} \tag{20}$$

where $M \doteq P^{-1}$ and $c \doteq P^{-1}x_c$. From a computational viewpoint, this second approach results to be more appealing. Indeed, using a slack variable ζ, it is possible to obtain the following system of $3n_\xi + 1$ linear inequalities

$$\begin{cases} m_i^T \xi - c_i & \leq \zeta_i, \quad i = 1, \ldots, n_\xi \\ -m_i^T \xi + c_i & \leq \zeta_i, \quad i = 1, \ldots, n_\xi \\ \sum_i^{n_\xi} \zeta_i & \leq 1. \end{cases}$$

The same convex optimization problem of (18) could be solved to define the center and the shape of the *largest* ℓ_∞-poly inscribed in $\mathbb{D}$. However, this would involve an exponential number of vertices 2^{n_ξ}. To avoid this, an approach based on Farkas lemma can be adopted, exploiting again a formulation in terms of linear inequalities. The details are not reported here due to space limitations. In this second case, once the center x_c and the rotation matrix P have been obtained, the corresponding $\mathcal{H}$-poly has only $2n_\xi$ hyperplanes, each one representing a different linear inequality.

[1] Note the difference under the $\log_2$ with respect to (8).

Once the initial SAS, $\mathbb{S}_S$ and the ℓ_1- and ℓ_∞-polys, i.e. $\mathbb{S}_1$ and $\mathbb{S}_\infty$ respectively, have been evaluated in terms of linear inequalities, the probabilistic scaling approach can be applied to determine the corresponding scaling factor γ. The scaling procedure is described in details in the paper [15]. For the sake of completeness, in the next subsection we recall its basic ideas and illustrate its application to the SAS case.

B. SAS probabilistic scaling

Given a candidate SAS set, the following simple algorithm can be used to guarantee with prescribed probability $1-\delta$ that the scaled set $\mathbb{S}(\gamma)$ is a good inner approximation of $\mathbb{X}_\varepsilon$.

Algorithm 1 Probabilistic SAS Scaling

Given probability levels ε and δ, let

$$N_\gamma \geq \frac{7.67}{\varepsilon} \ln \frac{1}{\delta} \text{ and } r = \left\lceil \frac{\varepsilon N_\gamma}{2} \right\rceil .$$

Draw N_γ samples of the uncertainty $q^{(1)}, \ldots, q^{(N_\gamma)}$

for $i = 1$ to N_γ **do**

Solve the optimization problem

$$\begin{aligned} \gamma_i \doteq & \arg\max \gamma \\ & \text{s.t.} \quad \mathbb{S}(\gamma) \subseteq \mathbb{X}_i \end{aligned} \tag{21}$$

end for

Return the r-th smallest value of γ_i.

The following Lemma applies to Algorithm 1.

Lemma 1: Given a candidate SAS set in the form $\mathbb{S}(\gamma) = x_c \oplus \gamma \mathbb{S}$, assume that $x_c \in \mathbb{X}_\varepsilon$. Then, Algorithm 1 guarantees that

$$\mathbb{S}(\gamma) \subseteq \mathbb{X}_\varepsilon$$

with probability at least $1-\delta$.

The proof of to Lemma 1 follows from Proposition 1 in [15] and Lemma 1 in [24].

IV. UAV CONTROL OVER A SLOPED VINEYARD

The selected application involves a fixed-wing UAV performing a monitoring mission over a Dolcetto vineyard at Carpeneto, Alessandria, Italy $(44°40'55.6''\text{N}, 8°37'28.1''\text{E})$. The Mission Planner of ArduPilot open source autopilot has been used to identify a grid pattern with a peculiar path orientation with respect to the grapevine rows, as shown in Fig. 2. The main objective is to provide proper control capabilities to the UAV to guarantee a fixed relative altitude with respect to the terrain of 150 m while following the desired optimal path defined by the guidance algorithm (described in detail in [25]) at a constant airspeed, i.e. $V_{ref} = 12$ m/s. The controllability of the aircraft shall be guaranteed despite the presence of external disturbances. For this application, the disturbance source is represented by a fixed-direction wind turbulence, which intensity can randomly vary among ± 1 m/s.

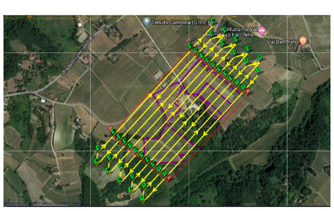

Fig. 1. Carpeneto vineyard, Piedmont, Italy (credit: Google).

TABLE I

STATE AND INPUT CONSTRAINTS BOUNDARIES.

State variable	Boundary	Input variable	Boundary
u [m/s]	± 1.5	ΔT [-]	± 0.36
h [rad]	± 2	δ_e [deg]	± 20

For validation purpose, the longitudinal control of the UAV has been analyzed, exploiting both OS-SMPC and the new PS-SMPC. For this scenario, the controlled state variables are the longitudinal component of the total airspeed in body axes u and the altitude h whereas the control variables are represented by the throttle command ΔT and the elevator deflection δ_e. The complete linearized model for the aircraft dynamics can be found in [26]. The main objective is to track reference airspeed and altitude profiles, defined by the grid path, during a monitoring mission to guarantee that UAV velocity u and quote h remain within predefined boundaries, applying optimal control actions, also bounded by physical limitations, as defined in Table I. In compliance with typical setting for MPC applied to UAV (see e.g. [14]), the prediction horizon T has been set equal to 15. Consequently, with $\varepsilon = 0.05$, $\delta = 10^{-6}$, we get $N_{LT} = 20,604$ and $N_\gamma = 2,063$. On the other hand, the sample complexity selected for generating the ℓ_1-poly has been set equal to $N_D = 100$ obtaining $3,500$ hyper-planes but only 107 linear constraints implemented online.

The preliminary results are represented in Fig. 2 and Fig. 3. We can notice that both MPC schemes provide acceptable tracking capabilities, despite larger (but still acceptable) deviations from the reference trajectory can be observed when the scaled set is exploited. Moreover, from Fig. 3, we can observe that both controllers are able to track the airspeed signal maintaining the UAV velocity within the boundaries. On the other hand, the vertical component of the wind significantly affects the aircraft altitude, which in some cases violates the given tracking constraint, deviating over 2 m from the reference path. This behavior has been observed for both controllers during the turn phases and it can be ascribable to neglected nonlinearities.

Additional details on controller performance are reported in Tab. II in terms of maximum and average values of the computational time required to solve *online* the SMPC problem for 3 different run each. The results show a significant reduction (about 100 times lower) of the computational load when a lower complexity constraint set is employed. This makes the PS-SMPC approach not only effective from a performance viewpoint but also presumably compliant with the computational capabilities of typical autopilot hardware.

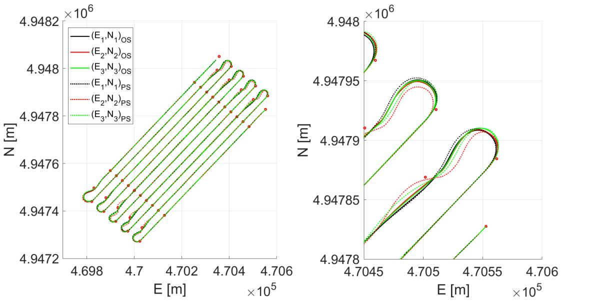

Fig. 2. UAV controlled trajectories obtained running OS-SMPC and PS-SMPC three times each.

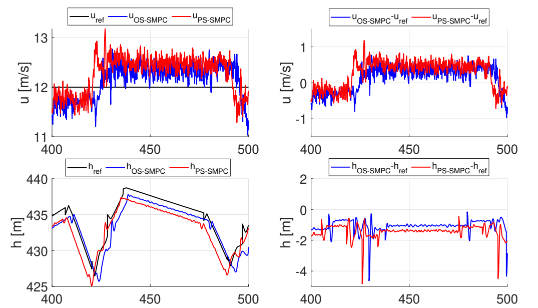

Fig. 3. Zoom-in on the behavior of controlled state variables with respect to corresponding reference signals (black lines) during a turn phase around two consecutive WPs (at $t = 420$ s and $t = 490$ s).

TABLE II

MAXIMUM AND AVERAGE ONLINE COMPUTATIONAL COST.

n.	$t_{c_{MAX_{OS}}}$	$t_{c_{AVG_{OS}}}$	$t_{c_{MAX_{PS}}}$	$t_{c_{AVG_{PS}}}$
1	2.0959	0.4178	0.0966	0.0087
2	2.9411	0.5626	0.7221	0.0190
3	2.1497	0.5434	0.2628	0.0086

V. CONCLUSIONS

In this paper, we proposed a novel approach which exploits a probabilistic scaling technique recently proposed by some of the authors to derive a novel SMPC scheme. The introduced framework exhibits a lower computational complexity, while sharing the appealing probabilistic guarantees of offline sampling.

REFERENCES

[1] M. Farina, L. Giulioni, and R. Scattolini, "Stochastic linear model predictive control with chance constraints–a review," *Journal of Process Control*, vol. 44, pp. 53–67, 2016.

[2] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[3] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.

[4] J. M. Grosso, P. Velarde, C. Ocampo-Martinez, J. M. Maestre, and V. Puig, "Stochastic model predictive control approaches applied to drinking water networks," *Optimal Control Applications and Methods*, vol. 38, no. 4, pp. 541–558, 2017.

[5] H. A. Nasir, A. Carè, and E. Weyer, "A randomised approach to flood control using value-at-risk," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 3939–3944.

[6] D. Van Hessem and O. Bosgra, "Stochastic closed-loop model predictive control of continuous nonlinear chemical processes," *Journal of Process Control*, vol. 16, no. 3, pp. 225–241, 2006.

[7] R. M. Vignali, F. Borghesan, L. Piroddi, M. Strelec, and M. Prandini, "Energy management of a building cooling system with thermal storage: An approximate dynamic programming solution," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 619–633, 2017.

[8] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Stochastic MPC with offline uncertainty sampling," *Automatica*, vol. 81, no. 1, pp. 176–183, 2017.

[9] M. Mammarella, M. Lorenzen, E. Capello, H. Park, F. Dabbene, G. Guglieri, M. Romano, and F. Allgöwer, "An offline-sampling SMPC framework with application to autonomous space maneuvers," *IEEE Transactions on Control Systems Technology*, pp. 1–15, 2018.

[10] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system," in *Robot Operating System (ROS)*. Springer, 2017, pp. 3–39.

[11] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Robust model predictive flight control of unmanned rotorcrafts," *Journal of Intelligent & Robotic Systems*, vol. 81, no. 3-4, pp. 443–469, 2016.

[12] T. J. Stastny, A. Dash, and R. Siegwart, "Nonlinear mpc for fixed-wing UAV trajectory tracking: Implementation and flight experiments," in *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1512.

[13] N. Michel, S. Bertrand, G. Valmorbida, S. Olaru, and D. Dumur, "Design and parameter tuning of a robust model predictive controller for UAVs," in *2017 20th IFAC World Congress*, 2017.

[14] M. Mammarella, E. Capello, F. Dabbene, and G. Guglieri, "Sample-based SMPC for tracking control of fixed-wing UAV," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 611–616, 2018.

[15] T. Alamo, V. Mirasierra, F. Dabbene, and M. Lorenzen, "Safe approximations of chance constrained sets by probabilistic scaling," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 1380–1385.

[16] F. Dabbene, C. Lagoa, and P. Shcherbakov, "On the complexity of randomized approximations of nonconvex sets," in *2010 IEEE International Symposium on Computer-Aided Control System Design*. IEEE, 2010, pp. 1564–1569.

[17] G. Sylvester, G. Rambaldi, D. Guerin, A. Wisniewski, N. Khan, J. Veale, and M. Xiao, "E-agriculture in action-drones for agriculture. food and agriculture organization of the united nations and international telecommunication union," *FAO of the UN and ITU*, vol. 19, 2018.

[18] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Constraint-tightening and stability in stochastic model predictive control," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3165–3177, 2017.

[19] M. Vidyasagar, *Learning and Generalisation: with Applications to Neural Networks*. Springer Science & Business Media, 2013.

[20] T. Alamo, R. Tempo, and E. F. Camacho, "Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2545–2559, 2009.

[21] S. Yan, P. Goulart, and M. Cannon, "Stochastic model predictive control with discounted probabilistic constraints," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 1003–1008.

[22] L. Hewing and M. N. Zeilinger, "Stochastic model predictive control for linear systems using probabilistic reachable sets," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 5182–5188.

[23] V. Kaibel and M. E. Pfetsch, "Some algorithmic problems in polytope theory," in *Algebra, geometry and software systems*. Springer, 2003, pp. 23–47.

[24] S. Mammarella, T. Alamo, F. Dabbene, and M. Lorenzen, "Computationally efficient stochastic MPC: a probabilistic scaling approach," in *arXiv 2005.10572 (eess.SY)*, 2020, pp. 1–8.

[25] M. Mammarella, G. Ristorto, E. Capello, N. Bloise, G. Guglieri, and F. Dabbene, "Waypoint tracking via tube-based robust model predictive control for crop monitoring with fixed-wing UAVs," in *2019 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*. IEEE, 2019, pp. 19–24.

[26] M. Mammarella and E. Capello, "Tube-based robust mpc processor-in-the-loop validation for fixed-wing uavs," *Journal of Intelligent & Robotic Systems*, pp. 1–20, 2020.