An Offline-Sampling SMPC Framework with Application to Autonomous Space Maneuvers

Martina Mammarella[1] *Member, IEEE*, Matthias Lorenzen[2], Elisa Capello[3] *Member, IEEE*, Hyeongjun Park[4] *Member, IEEE*, Fabrizio Dabbene[5], *Senior Member, IEEE*, Giorgio Guglieri[1], Marcello Romano[4], *Senior Member, IEEE*, and Frank Allgöwer[2], *Member, IEEE*

***Abstract*—In this paper, a sampling-based Stochastic Model Predictive Control algorithm is proposed for discrete-time linear systems subject to both parametric uncertainties and additive disturbances. One of the main drivers for the development of the proposed control strategy is the need of reliable and robust guidance and control strategies for automated rendezvous and proximity operations between spacecraft. To this end, the proposed control algorithm is validated on a floating spacecraft experimental testbed, proving that this solution is effectively implementable in real time. Parametric uncertainties due to the mass variations during operations, linearization errors, and disturbances due to external space environment are simultaneously considered. The approach enables to suitably tighten the constraints to guarantee robust recursive feasibility when bounds on the uncertain variables are provided. Moreover, the offline sampling approach in the control design phase shifts all the intensive computations to the offline phase, thus greatly reducing the online computational cost, which usually constitutes the main limit for the adoption of Stochastic Model Predictive Control schemes, especially for low-cost on-board hardware. Numerical simulations and experiments show that the approach provides probabilistic guarantees on the success of the mission, even in rather uncertain and noisy situations, while improving the spacecraft performance in terms of fuel consumption.**

***Index Terms*— Stochastic Model Predictive Control; Chance Constraints; Sampling-based Approach; Real-time Implementability; Autonomous Rendezvous between Spacecraft.**

I. Introduction

In the last decades, model predictive control (MPC) has become one of the most successful advanced control techniques for industrial processes, thanks to its ability to handle multi-variable systems, explicitly taking into account state and equipment constraints, see for instance the recent survey [1].

[1]M. Mammarella is with the Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy, martina.mammarella@polito.it

[2]M. Lorenzen and F. Allgöwer are with the Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany, (matthias.lorenzen, frank.allgower@ist.uni-stuttgart.de)

[3] E. Capello and G. Guglieri are with the Department of Mechanical and Aerospace Engineering and with the CNR–IEIIT, Politecnico di Torino, Torino, Italy, (elisa.capello,giorgio.guglieri@polito.it)

[4]H. Park is with the Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM, USA, hjpark@nmsu.edu

[5]F. Dabbene is with the CNR–IEIIT, Politecnico di Torino, Torino, Italy, fabrizio.dabbene@ieiit.cnr.it

[6]M. Romano is with the Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA, USA, mromano@nps.edu

Early publications on the topic already emphasized that moving horizon schemes like MPC might incur significant performance degradation in the presence of uncertainty [2]. Furthermore, ignoring modeling errors and disturbances can lead to constraint violation in closed loop and the online optimization being infeasible. To cope with these disadvantages, in the last years Robust MPC has received a great deal of attention and, at least for linear systems, it can nowadays be considered well-understood and having achieved a mature state [3]. More recently, for processes where a stochastic model can be formulated to represent the uncertainty and disturbance and constraints violation does not correspond to compromise the application or lose the mission, Stochastic Model Predictive Control (SMPC) approaches have gained popularity [4]. Indeed, a probabilistic model allows to optimize average performance or appropriate risk measures, and the introduction of so-called *chance constraints*, which seem more appropriate in those applications where allowing a (small) probability of constraints violation provides a higher cost-effectiveness of the application itself. Furthermore, chance constraints lead to an increased region of attraction and enlarge the set of states for which MPC provides a valid control law [5].

On the other hand, the classical criticism of MPC schemes, especially in their robust/stochastic instantiations, is their *slowness*. This has limited their application to problems involving slow dynamics, where the sample time is measured in tens of seconds or minutes. In particular, due to the increased computational load, SMPC has mainly been applied for: (i) slow systems, as e.g. water networks [6], river flood control [7] and chemical processes [8], and (ii) fast-dynamics systems in which a dedicated high-performance computing platform is exploited to solve online the optimization problem, e.g. [9], in which an NVIDIA Tesla k20 GPU allows to run parallelized code efficiently.

Indeed, typically this widely recognized shortcoming is mainly due to the computational effort required in the online solution of the ensuing optimization problem, and to the difficulty of embedding a real-time solver for MPC implementation. For example, in [10], a heuristic Randomized MPC is proposed and experimentally validated for controlling mini race cars. In this case, the optimization problem involves a small number of samples (lower than 100) and is solved on a desktop PC which streams the optimal control input to the cars via bluetooth. Indeed, when the number of samples, variables and/or prediction horizon increases and the system

to be controlled is also characterized by fast dynamics (in the order of ms or s), the problem could easily become intractable. A practical solution proposed in the literature is to evaluate offline the control law, and then the control action is implemented online as a lookup table [11]. However, this solution renders the controller less apt to deal with model uncertainties and external disturbances. Moreover, a substantial computational capability and large memory are required, especially for systems with fast dynamics, such as UAV, aircraft, and spacecraft.

In space applications, the available processors provide limited on board computational power. In this framework, the requirement of real-time implementability for new Guidance Navigation and Control (GNC) algorithms gains the highest priority.

The contribution of the paper is twofold. From a theoretical viewpoint, the paper integrates and extends the previous works of the authors [5], [12], proposing offline sample-based strategies for addressing in a computationally tractable manner SMPC. In particular, as detailed Section I-A, the paper develops for the first time a complete and integrated framework, able to cope simultaneously with additive random noise and parametric stochastic uncertainty.

From an application viewpoint, the paper demonstrates real-time implementability of the proposed scheme, addressing a very important control problem arising in aerospace applications, the Autonomous Rendezvous and Docking (ARVD) maneuver among spacecraft. Indeed, as discussed in Section I-B, the ability to carry over proximity operations in a completely autonomous manner represents one of the main challenges of modern spacecraft missions. These require the capability of dealing in an efficient way with external disturbances due to the space environment, and with uncertainties. The SMPC scheme is shown to be able to cope with all these requirements, providing sufficiently high guarantees in terms of safety and constraints satisfaction, and at the same time being sufficiently fast to be implemented in a real-time framework.

The framework developed in this paper has recently been applied in [13] to the problem of guidance and control of a fixed-wing UAV for urban monitoring applications. In that paper, it is shown how the approach can be extended to tracking, and can easily deal with the time-constants involved in UAV dynamics.

In the next section, we highlight the contributions of the present work to the SMPC theory, while the next section describes in detail the application example considered, highlighting how it can benefit from the performance guarantees provided by the introduced control framework.

A. A Novel Stochastic Model Predictive Control Framework

The main problem encountered in the design of SMPC algorithms is the derivation of computationally tractable methods to propagate the uncertainty for evaluating the cost function and the chance constraints. Both problems involve multivariate integrals, whose evaluation requires the development of suitable techniques. An exact evaluation is in general only possible for linear systems with additive Gaussian disturbance, where the constraints can be reformulated as second-order cone constraints [14], or for finitely supported disturbances as in [15]. Approximate solutions include a particle approach [16] or polynomial chaos expansion [17]. Among the different methods, randomized algorithms [18], and in particular the scenario approach [19], represent one of the most promising one when the computational capability of the stochastic algorithm represents a stringent constraint for the chosen application and several types of stochastic uncertainty are involved and can enter nonlinearly in the system. The first approaches in this direction can be found in the methods proposed in [20], [21], [22], in which the uncertainty is propagated as a finite number of scenarios to be considered at each step. However, these approaches may still be rather demanding for real-time implementations, since different samples need to be drawn at each step. Recently, this drawback was overcome by the introduction of *offline sampling* strategies, that allow to reduce the computational effort made online by means of a pre-processing of data made offline. In particular, in [12] this approach was developed for problems involving additive disturbances, acting on a nominal system. In [5], parametric uncertainties are instead considered in a noise-free setting.

This paper solves the nontrivial problem of extending the previous result into a comprehensive framework, able to tackle situations in which *both* additive disturbances and parametric uncertainties are simultaneously present. The SMPC scheme here proposed results to be less conservative than most analytical approaches based on Chebyschev's inequality (see [23] and [24] for further details), is computationally tractable and guarantees recursive feasibility. As in [12], the computational load is reduced by generating scenarios offline and keeping only selected, *necessary* samples for the online optimization. The algorithm guarantees robust satisfaction of the input constraints and bounds on the confidence that the chance constraints are satisfied can be chosen by the designer. Due to the additive disturbance, the state does not converge to the origin. Instead, an asymptotic performance bound is provided. The presented theory is attractive for real-world applications, since the design can be based on real data gathered from experiments or high fidelity simulations. Moreover, thanks to the offline sampling approach, this SMPC scheme can be applied to relatively fast dynamics, as those of space platforms during the final phase of the automated rendezvous and mating maneuver.

B. ARVD Problem (Problem Setup)

The advancement of robotics and autonomous systems will be central to the transition of space missions from current ground-in-the-loop (geocentric) architectures to more autonomous systems. For instance, the Committee on NASA Technology Roadmaps has highlighted as "Robotics, Tele-Robotics, and Autonomous Systems" shall be regarded as high-priority technology area in broadening access to space and expanding human presence in the Solar System [25]. Among them, ARVD represents the cornerstone technology, since all of the scenarios that space agencies have defined for the future exploration program have one thing in common:

each mission architecture heavily relies on the ability to rendezvous and mate multiple elements in space autonomously. In order to meet the exploration enterprise goals of affordability, safety and sustainability, the critical capabilities of rendezvous, capture and in-space assembly must become autonomous, increasing their reliability [26]. The complexity of the ARVD mainly results from the multitude of safety and operational *constraints* which must be fulfilled. These constraints are defined with respect to the rendezvous approach phase considered. In terms of safety, the close range rendezvous phase is the most critical, since the space systems involved are relatively close together and the trajectory of the chaser, by definition, leads toward the target, so that any deviation from the planned trajectory can potentially lead to a collision. Therefore, the main focus of this paper is on the final approach between the chaser vehicle and the target one, considering the typical minus V-bar approach.

First, for sensing purposes (see [27]), it is required that the chaser vehicle remains inside a Line-Of-Sight (LOS) from the docking point, constraint usually defined in terms of an approaching corridor, as represented in Figure 1, which can be modeled as a polytope (without any generality loss, a rectangular parallelepiped can be used). Moreover, soft docking constraints can be enforced, reducing the approach velocity in line with distance to the target, as well as limiting the maximum approach velocity. When using thrusters for spacecraft trajectory control, not only there are constraints on the maximum force that can be applied at any given instant, i.e. saturation of the actuators, but there is also the physical constraint of a thrust "dead-zone" between the thruster being fully off, and delivering its minimum non-zero thrust, often referred to as the "Minimum Impulse Bit" (MIB). Indeed, constraints on the maximum deliverable Δv are placed on each element of the input vector. Last but not least, another constraint can be imposed on the fuel consumption or on the amount of fuel dedicated to the maneuver.

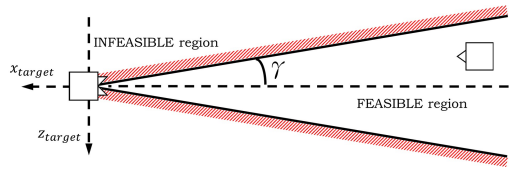


Fig. 1. Line-Of-Sight constraint defined in terms of infeasible/feasible region considering a minus V-bar approach [28].

A second important challenge for close-range ARVD is represented by the need to handle *uncertainty*. Thruster firings, mass and geometry model, aerodynamic drag in Low Earth Orbit (LEO), Inertial Measurement Unit (IMU) and camera measurements introduce uncertainties in relative state knowledge and control accuracy. As the chaser spacecraft approaches its target, these uncertainties can induce violations in any of the aforementioned mission constraints. Hence, one should embed in ARVD algorithms the capability to handle any expected uncertainty directly, i.e. incorporating strategies to deal with all known unknowns. The key for ARVD GNC strategies is relying on solution techniques that can be made efficient for real-time implementation. Indeed, to meet the GNC challenges of next-generation space missions, onboard algorithms will need to satisfy the following specifications: (i) *real-time implementability*; (ii) *optimality*; (iii) *verifiability*. Therefore, new GNC algorithms need to be implemented and executed on real-time processors, in a compatible amount of time, providing a feasible and (approximately) optimal solution, verifying the design metrics identified to describe their performance.

Several methodologies have been proposed in literature for ARVD, which have shown robustness with respect to known and unknown uncertainty and disturbance affecting the system during the final phase of the rendezvous maneuver. The reader is referred to [29] for a recent survey. In particular, we want to recall the approach proposed in [30], where a robust MPC is adopted to solve the problem of spacecraft rendezvous, using the Hill-Clohessy-Wiltshire (HCW) model and including additive disturbances and LOS constraints. Furthermore, it has been proved that a robust approach implies higher fuel consumption with respect to classical methods where disturbances are neglected (see [31]). On the other hand, still a probability of constraints violation can be considered thanks to the possibility of relaxing the safety requirements also during the critical final phase of ARVD. In this work, a stochastic approach is proposed in order to relax the safety trajectory constraints reducing the conservativeness with respect to a robust approach, as well as fuel consumption, optimizing the average performance and allowing an affordable level of constraints violation.

The remainder of this paper is organized as follows. Section II introduces the finite horizon receding optimal control problem, starting with a suitable formulation of the constraints through an offline uncertainty sampling approach. Thereafter, the SMPC scheme algorithm is introduced, and its main theoretical properties are summarized and proved. In Section III, the experimental testbed used to validate the real-time implementability of the proposed scheme is described. Its dynamic model is derived, including the identification and modeling of uncertainty and additive disturbance and presenting the main issues linked to real-time implementability and principal solvers investigated. The simulation and experimental results are discussed in Section IV and the algorithm performances are discussed with respect to computational effort and fuel consumption. Finally, Section V provides some conclusions and directions for future works.

Notation: Uppercase letters are used for matrices and lower case for vectors. $[A]_j$ and $[a]_j$ denote the j-th row and entry of the matrix A and vector a, respectively. Positive (semi)definite matrices A are denoted $A \succ 0$ ($A \succeq 0$) and $\|x\|_A^2 = x^T A x$. The set $\mathbb{N}_{>0}$ denotes the positive integers and $\mathbb{N}_{\geq 0} = \{0\} \cup \mathbb{N}_{>0}$, similarly $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$ for positive real numbers. The notation $\mathbb{P}_k\{\mathcal{A}\} = \mathbb{P}\{\mathcal{A}|x_k\}$ denotes the conditional probability of an event $\mathcal{A}$ given the realization of x_k, similarly $\mathbb{E}_k\{\mathcal{A}\} = \mathbb{E}\{\mathcal{A}|x_k\}$ for the expected value. We use x_k for the (measured) state at time k and $x_{\ell|k}$ for the state predicted l steps ahead at

time k. The sequence of length T of vectors $v_{0|k}, \dots, v_{T|k}$ is denoted by $\mathbf{v}_{T|k}$. $A \oplus B = \{a+b |\, a \in A, b \in B\}$, $A \ominus B = \{a \in A |\, a+b \in A, \forall b \in B\}$ denote the Minkowski sum and the Pontryagin set difference, respectively. The notation $\mathbf{1}_n$ is used to denote a n-dimensional vector of ones.

II. SMPC DESIGN UNDER UNCERTAINTY AND RANDOM NOISE

We consider the following discrete-time system subject to both random noise and stochastic uncertainty

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

with state $x_k \in \mathbb{R}^n$, control input $u_k \in \mathbb{R}^m$, additive disturbance $w_k \in \mathbb{R}^{m_w}$, and parametric uncertainty $q_k \in \mathbb{R}^{n_q}$.

In the following, the disturbance sequence $(w_k)_{k\in\mathbb{N}_{\geq 0}}$ is assumed to be a realization of a stochastic process $(W_k)_{k\in\mathbb{N}_{\geq 0}}$.

Assumption 1 (Bounded Random Disturbance)**.** *The disturbances W_k, for $k = 0, 1, 2...$, are independent and identically distributed (iid), zero-mean random variables with support $\mathbb{W}$, which is a bounded and convex set.*

We assume that the system matrices $A(q_k)$, $B(q_k)$, and $B_w(q_k)$, of appropriate dimensions, are (possibly nonlinear) functions of the uncertainty q_k. The uncertainty vector q_k belongs to a given set $\mathbb{Q}$ and satisfies the following assumption.

Assumption 2 (Stochastic Uncertainty)**.** *The uncertainties $q_k \in \mathbb{Q} \subset \mathbb{R}^{n_q}$, for $k \in \mathbb{N}$, are realizations of i.i.d. multivariate real valued random variables Q_k. Moreover, let $\mathbb{G} = \{(A(q_k), B(q_k), B_w(q_k)\}_{q_k\in\mathbb{Q}}$, a polytopic outer approximation with N_c vertexes $\bar{\mathbb{G}} \doteq co\left\{A^j, B^j, B_w^j\right\}_{j\in\mathbb{N}_1^{N_c}} \supseteq \mathbb{G}$ exists and is known.*

The system is subject to p_x individual chance-constraints on the state and m hard constraints on the input

$$\mathbb{P}\left\{[H_x]_\alpha x_{\ell|k} \leq [h_x]_\alpha\right\} \geq 1-\varepsilon_\alpha, \forall\, \ell \in \mathbb{N}_{\geq 0},\ \alpha \in \mathbb{N}_1^{p_x} \tag{2a}$$

$$H_u u_{\ell|k} \leq h_u, \forall\, \ell \in \mathbb{N}_{>0}, \tag{2b}$$

with $H_x \in \mathbb{R}^{p_x\times n}$, $h_x \in \mathbb{R}^{p_x}$, $H_u \in \mathbb{R}^{m\times m}$, $h_x \in \mathbb{R}^m$, and $\varepsilon_\alpha \in (0,1)$. Note that the probability $\mathbb{P}$ in (2) denotes the joint probability with respect to both, the uncertainty and disturbance sequences $\mathbf{q}_k$ and $\mathbf{w}_k$, where $\mathbf{q}_k \doteq \left\{q_{\ell|k}\right\}_{\ell\in\mathbb{N}_0^{T-1}}$ and $\mathbf{w}_k \doteq \left\{w_{\ell|k}\right\}_{\ell\in\mathbb{N}_0^{T-1}}$. Then, as typical in stabilizing MPC, we assume that a suitable terminal set $\mathbb{X}_T$ and an asymptotically stabilizing control gain for (1) exist.

Assumption 3 (Terminal set)**.** *There exists a terminal set $\mathbb{X}_T = \{x_k\,|H_T x_k \leq h_T\}$, which is robustly forward invariant for (1) under the (given) control law $u_k = Kx_k$. Given any $x_k \in \mathbb{X}_T$, the state and input constraints (2) are satisfied and there exists $P \in \mathbb{R}^{n\times n}$ such that*

$$Q + K^T R K + \mathbb{E}[A_{cl}(q_k)^T P A_{cl}(q_k)] - P \preceq 0 \tag{3}$$

for all $q \in \mathbb{Q}$, with $A_{cl}(q_k) \doteq A(q_k) + B(q_k)K$, and with $Q \in \mathbb{R}^{n\times n}$, $Q \succ 0$, $R \in \mathbb{R}^{m\times m}$, $R \succ 0$.

Following a scheme adopted in [4] for defining the predicted control sequence for nominal, robust and also stochastic MPC, we consider the design of a parametrized feedback policy of the form

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

where K satisfies is quadratically stabilizing for the system (1) and, for a given $x_{0|k} = x_k$, the sequence of correction terms $\mathbf{v}_k \doteq \left\{v_{\ell|k}\right\}_{\ell\in\mathbb{N}_0^{T-1}}$ is determined by the SMPC algorithm as the minimizer of the expected finite-horizon cost

$$J_T(x_k, \mathbf{v}_k) = \mathbb{E}\left\{\sum_{l=0}^{T-1}(x_{\ell|k}^T Q x_{\ell|k} + u_{\ell|k}^T R u_{\ell|k}) + x_{T|k}^T P x_{T|k}\right\}, \tag{5}$$

subject to constraints (2).

A. Offline Uncertainty Sampling for SMPC

For the following analysis, we first explicitly solve equation (1) with prestabilizing input (4) for the predicted states $x_{1|k}, \dots, x_{T|k}$ and predicted inputs $u_{0|k}, \dots, u_{T-1|k}$. In particular, simple algebraic manipulations show that it is possible to derive suitable transfer matrices $\Phi_{\ell|k}^0(\mathbf{q}_k)$, $\Phi_{\ell|k}^v(\mathbf{q}_k)$, $\Phi_{\ell|k}^w(\mathbf{q}_k)$, and Γ_ℓ (the reader is referred to Appendix A for details), such that

$$x_{\ell|k}(\mathbf{q}_k, \mathbf{w}_k) = \Phi_{\ell|k}^0(\mathbf{q}_k)x_k + \Phi_{\ell|k}^v(\mathbf{q}_k)\mathbf{v}_k + \Phi_{\ell|k}^w(\mathbf{q}_k)\mathbf{w}_k \tag{6a}$$

$$u_{\ell|k}(\mathbf{q}_k, \mathbf{w}_k) = K\Phi_{\ell|k}^0(\mathbf{q}_k)x_k + (K\Phi_{\ell|k}^v(\mathbf{q}_k) + \Gamma_\ell)\mathbf{v}_k + K\Phi_{\ell|k}^w(\mathbf{q}_k)\mathbf{w}_k, \tag{6b}$$

In the previous equations, we highlight that both predicted states and inputs are function of the uncertainty and the noise sequence, $\mathbf{q}_k$ and $\mathbf{w}_k$ respectively. Given the solution (6), the expected value of the finite-horizon cost (5) can be evaluated offline, leading to a quadratic cost function of the form

$$J_T(x_k, \mathbf{v}_k) = [x_k^T\ \mathbf{v}_k^T\ \mathbf{1}_{m_w}^T]\tilde{S}\begin{bmatrix} x_k \\ \mathbf{v}_k \\ \mathbf{1}_{m_w}\end{bmatrix} \tag{7}$$

in the deterministic variables x_k and $\mathbf{v}_k$. The evaluation of $\tilde{S}$ requires the computation of an expected value, which can be explicitly evaluated or sufficiently exact approximated taking random samples of the sequences $\mathbf{q}_k$ and $\mathbf{w}_k$ (see again Appendix A for details).

We now follow the same approach proposed in [12], and observe that an inner approximation for the chance constraint (2a) can be derived in the form of linear constraints on x_k, $\mathbf{v}_k$ and $\mathbf{w}_k$, utilizing a sampling-based approach. In particular, for each probabilistic state constraint $\alpha \in \mathbb{N}_1^{p_x}$, and for each time step $\ell \in \mathbb{N}_0^{T-1}$, let us define the corresponding chance-constrained set as follows

$$\mathbb{X}_\ell^{P,\alpha} = \left\{x_k, \mathbf{v}_k \;|\; \mathbb{P}\left\{[H_x]_\alpha x_{\ell|k}(\mathbf{q}_k, \mathbf{w}_k) \leq [h_x]_\alpha\right\} \geq 1-\varepsilon_\alpha\right\}. \tag{8}$$

In the above definition, we use the apex P as in [12] to indicate that the set has probabilistic nature. Then, exploiting results from statistical learning theory [32], an estimate of $\mathbb{X}_\ell^{P,\alpha}$ may be constructed extracting N_ℓ^x iid sample sequences $\mathbf{q}^{(i_\ell^x)}$ and $\mathbf{w}^{(i_\ell^x)}$, with $i_\ell^x \in \mathbb{N}_1^{N_\ell^x}$, and building the corresponding *sampled state constraint set*

$$\mathbb{X}_\ell^{S,\alpha} = \left\{ x_k, \mathbf{v}_k \mid [H_x]_\alpha x_{\ell|k}(\mathbf{q}^{(i_\ell^x)}, \mathbf{w}^{(i_\ell^x)}) \le [h_x]_\alpha, i_\ell^x \in \mathbb{N}_1^{N_\ell^x} \right\},$$

for $\ell \in \mathbb{N}_0^{T-1}$. The apex S is used to indicate that the set is the outcome of a sampling process.

In particular it was shown in [12] that, for given probabilistic levels $\delta \in (0,1)$ and $\varepsilon_\alpha \in (0, 0.14)$, if we define

$$\tilde{N}(d, \varepsilon_\alpha, \delta) = \frac{4.1}{\varepsilon_\alpha}\Big(\ln \frac{21.64}{\delta} + 4.39 d \log_2 \Big(\frac{8e}{\varepsilon_\alpha}\Big)\Big),$$

then the choice $N_\ell^x \ge \tilde{N}(p + \ell m, \varepsilon_\alpha, \delta)$ guarantees that $\mathbb{X}_\ell^{S,\alpha} \subseteq \mathbb{X}_\ell^{P,\alpha}$ with probability greater than $1-\delta$. Hence, we obtain that $x_{\ell|k} \in \mathbb{X}_\ell^{S,\alpha}$ is guaranteed with high probability whenever $x_{\ell|k}$ satisfies the following set of linear constraints

$$H_x x_{\ell|k}(\mathbf{q}^{(i_\ell^x)}, \mathbf{w}^{(i_\ell^x)}) \le h_x, \quad \text{for } i_\ell^x \in \mathbb{N}_1^{N_\ell^x}.$$

Note that, from (6a), the above equations rewrite as the following linear constraint in x_k, $\mathbf{v}_k$

$$\begin{bmatrix} \tilde{H}_x^x & \tilde{H}_x^u \end{bmatrix} \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \le \tilde{h}_x \tag{9}$$

where we defined

$$[\tilde{H}_x^x \;\; \tilde{H}_x^u] = \begin{bmatrix} H_x\Phi_{0|k}^0(\mathbf{q}^{(1)}) & H_x\Phi_{0|k}^v(\mathbf{q}^{(1)}) \\ \vdots & \vdots \\ H_x\Phi_{0|k}^0(\mathbf{q}^{(N_0^x)}) & H_x\Phi_{0|k}^v(\mathbf{q}^{(N_0^x)}) \\ \vdots & \vdots \\ H_x\Phi_{T-1|k}^0(\mathbf{q}^{(1)}) & H_x\Phi_{T-1|k}^v(q^{(1)}) \\ \vdots & \vdots \\ H_x\Phi_{T-1|k}^0(\mathbf{q}^{(N_{T-1}^x)}) & H_x\Phi_{T-1|k}^v(\mathbf{q}^{(N_{T-1}^x)}) \end{bmatrix}, \tag{10a}$$

$$\tilde{h}_x = \begin{bmatrix} h_x - H_x\Phi_{0|k}^w(\mathbf{q}^{(1)})\mathbf{w}_k^{(1)} \\ \vdots \\ h_x - H_x\Phi_{0|k}^w(\mathbf{q}^{(N_0^x)})\mathbf{w}_k^{(N_0^x)} \\ \vdots \\ h_x - H_x\Phi_{T-1|k}^w(\mathbf{q}^{(1)})\mathbf{w}_k^{(1)} \\ \vdots \\ h_x - H_x\Phi_{T-1|k}^w(\mathbf{q}^{(N_{T-1}^x)})\mathbf{w}_k^{(N_{T-1}^x)} \end{bmatrix}. \tag{10b}$$

Note that the total number of samples to be drawn to construct the sampled constraint sets (9) is equal to $N^x \doteq \sum_{\ell=0}^{T-1} N_\ell^x$, and thus the total number of linear inequalities will be pN^x. On the other hand, these sets can be be computed *offline*.

In a similar way, the hard input constraints can be approximated by introducing a suitable sampled approximation. To this end, for given probabilistic level $\varepsilon_\beta \in (0, 0.14)$ for each $\beta \in \mathbb{N}_1^{p_u}$, we draw $N_\ell^u \ge \tilde{N}(n + \ell m, \varepsilon_\beta, \delta)$ random samples and construct the *sampled input constraint set*

$$\mathbb{U}_\ell^{S,\beta} = \left\{ x_k, \mathbf{v}_k \mid [H_u]_\beta u_{\ell|k}(\mathbf{q}^{i_u}, \mathbf{w}^{(i_u)}) \le h_u, \quad i_u \in \mathbb{N}_1^{N_\ell^u} \right\}$$

for $\ell \in \mathbb{N}_0^{T-1}$, thus obtaining the N_ℓ^u linear constraints

$$H_u u_{\ell|k}(\mathbf{q}^{(i_u)}, \mathbf{w}^{(i_u)}) \le h_u,$$

which, from (6b), rewrites as the following linear constraint in x_k, $\mathbf{v}_k$

$$\begin{bmatrix} \tilde{H}_u^x & \tilde{H}_u^u \end{bmatrix} \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \le \tilde{h}_u. \tag{11}$$

where $\tilde{H}_u^x$ and $\tilde{H}_u^u$ are defined analogously to (10), and involve $N^u \doteq \sum_{\ell=1}^{T-1} N_\ell^u$ samples. Finally, for each $\gamma \in \mathbb{N}_1^n$, $\varepsilon_\gamma \in (0, 0.14)$, the terminal constraints can also be approximated by drawing $N_T \ge \tilde{N}(n + Tm, \varepsilon_\gamma, \delta)$ random samples and constructing the sets

$$\mathbb{X}_T^{S,\gamma} = \left\{ x_k, \mathbf{v}_k \mid [H_T]_\gamma x_{T|k}(\mathbf{q}^{i_T}, \mathbf{w}^{(i_T)}) \le h_T, \quad i_T \in \mathbb{N}_1^{N_T} \right\}$$

for $i_T \in \mathbb{N}_1^{N_T}$, which lead to

$$H_T x_{T|k}(\mathbf{q}^{(i_T)}, \mathbf{w}^{(i_T)}) \le h_T$$

that through (6a),

$$\begin{bmatrix} \tilde{H}_T^x & \tilde{H}_T^u \end{bmatrix} \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \le \tilde{h}_T \tag{12}$$

where $\tilde{H}_T^x$ and $\tilde{H}_T^u$ involve N^T samples.

The linear constraints (9), (11), (12), possibly after constraint reduction, can be summarized in the following linear constraint set

$$\begin{aligned} \mathbb{D} &= \left\{ x_k, \mathbf{v}_k \;\mid\; \begin{bmatrix} \tilde{H}_x^x & \tilde{H}_x^u \\ \tilde{H}_T^x & \tilde{H}_T^u \\ \tilde{H}_u^x & \tilde{H}_u^u \end{bmatrix} \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \le \begin{bmatrix} \tilde{h}_x \\ \tilde{h}_T \\ \tilde{h}_u \end{bmatrix} \right\} \\ &= \left\{ x_k, \mathbf{v}_k \;\mid\; \tilde{H} \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \le \tilde{h} \right\}. \end{aligned} \tag{13}$$

We note also that, due to the sampling procedure, these linear constraints are in general highly redundant. To cope with this issue, suitable algorithms for redundant constraints removal may be applied and the sets can be further simplified. The reader is referred to [12] for a thorough discussion on this issue.

Moreover, again similar to [12], a first step constraint is added to (13), defined starting from the set

$$\mathbb{C}_T = \left\{ \begin{bmatrix} x_k \\ v_{0|k} \end{bmatrix} \in \mathbb{R}^{n+m} \;\middle|\; \begin{array}{c} \exists v_{1|k}, \cdots, v_{T-1|k} \in \mathbb{R}^n, \\ s.t.\ (x_k, \mathbf{v}_k) \in \mathbb{D} \end{array} \right\} \tag{14}$$

which defines the set of feasible states and first inputs of the scenario program with given fixed samples. Therefore, we can define $\mathbb{C}_{T,x}^\infty = \{x_k \mid H_\infty x_k \le h_\infty\}$ as the (maximal) robust control invariant set for the system (1) with $(x_k, u_k) \in \mathbb{C}_T$. Finally, in order to ensure robust recursive feasibility, a constraint on the first input is added to (13) and the additional constraint set is given by

$$\begin{aligned} \mathbb{D}_R = \Big\{ x_k, \mathbf{v}_k \;\mid\; & H_\infty A_{cl}^j x_k + H_\infty B^j v_{0|k} \le \\ & h_\infty - H_\infty B_w^j w_{0|k} \Big\} \end{aligned} \tag{15}$$

with A^j, B^j, B^j_w from Assumption 2 and $A^j_{cl} = A^j + B^j K$. The final set of linear constraints to be employed in online implementation is thus given by the intersection of the sets $\mathbb{D}$ and $\mathbb{D}_R$, defined in (13) and (15) respectively.

B. SMPC Algorithm Based on Offline Sampling

The complete sampling-based SMPC algorithm we propose is split into two parts: (i) an offline step, which comprises the sample generation and the computation of the ensuing sets, and (ii) a repeated online optimization. While the first step may be rather costly, the online implementation only involves the solution of quadratic programs, which may be carried out in a very efficient way. A detailed description of the Offline Sampling-Based Stochastic Model Predictive Control (OS-SMPC) scheme is reported next.

OS-SMPC scheme

OFFLINE STEP. *Before running the online control algorithm:*

1) *Compute the finite-horizon cost matrix $\tilde{S}$ in (7);*
2) *Draw a sufficiently large number of samples to determine the sampled constraints $\mathbb{X}_\ell^{S,\alpha}$, $\mathbb{U}_\ell^{S,\beta}$, and $\mathbb{X}_T^{S,\gamma}$, defined respectively in (9), (11), (12),*
3) *Remove redundant constraints and get $\mathbb{D}$ in (13)*
4) *Determine the first step constraint set $\mathbb{D}_\mathbb{R}$ in (15).*

ONLINE IMPLEMENTATION. *At each time step k:*

1) *Measure the current state x_k;*
2) *Determine the minimizer of the quadratic cost (7) subject to the pre-computed linear constraints $\mathbb{D}$ and $\mathbb{D}_\mathbb{R}$*

$$\mathbf{v}_k^* = \arg \min_{\mathbf{v}_k} [x_k \ \mathbf{v}_k \ \mathbf{1}_{m_w}] \tilde{S} \begin{bmatrix} x_k \\ \mathbf{v}_k \\ \mathbf{1}_{m_w} \end{bmatrix} \tag{16a}$$

$$\text{s.t. } (x_k, \mathbf{v}_k) \in \mathbb{D} \cap \mathbb{D}_R; \tag{16b}$$

3) *Apply the control input*

$$u_k = K x_k + v_{0|k}^*,$$

where $v_{0|k}^*$ is the first control action of the optimal sequence $\boldsymbol{v}_k^*$.

In the next section, we prove several important properties of the proposed OS-SMPC scheme.

C. Theoretical Guarantees of OS-SMPC

First, we show that the introduction of the first step constraint $\mathbb{D}_R$ allows to prove recursive feasibility of the OS-SMPC scheme.

Proposition 1 (Recursive Feasibility)**.** *The closed loop dynamics $x_{k+1} = A_{cl}(q_k)x_k + B(q_k)v_k + B_w(q_k)w_k$ under the control law (4) renders the set $\mathbb{D} \cap \mathbb{D}_R$ forward invariant. In particular, let $\mathbb{V}(x_k) = \{\boldsymbol{v}_k \in \mathbb{R}^{mT} \mid (x_k, \boldsymbol{v}_k) \in \mathbb{D} \cap \mathbb{D}_R\}$. If $\boldsymbol{v}_k \in \mathbb{V}(x_k)$, then, for every realization q_k and w_k, and $x_{k+1} = A_{cl}(q_k)x_k + B(q_k)v_{0|k} + B_w(q_k)w_{0|k}$, the OS-SMPC guarantees*

$$\mathbb{V}(x_{k+1}) \neq \emptyset.$$

The proof follows similar lines to the one provided in [12], and is briefly sketched here: From $(x_k, \mathbf{v}_k) \in \mathbb{D}_R$ it follows $x_{k+1} \in \mathbb{C}^\infty_{T,x}$ robustly. Then, $\mathbb{C}^\infty_{T,x} \subset \{x \mid \mathbb{V}(x) \neq \emptyset\}$, by construction, which proves the claim.

The previous proposition, besides showing how the OS-SMPC algorithm guarantees recursive feasibility, it is also instrumental in proving that the control input returned by the algorithm guarantees satisfaction of the chance-constraints on the state and hard constraints on the input defined in (2). This is formally stated next.

Proposition 2 (Constraint Satisfaction)**.** *If $x_0 \in \mathbb{C}^\infty_{T,x}$, then the closed-loop system under the OS-SMPC control law, for all $k \geq 1$, satisfies each probabilistic state constraint (2a) with confidence $(1-\delta)$, and the hard input constraint (2b) robustly.*

Proof Since the OS-SMPC algorithm is robustly recursively feasible (Proposition 1), hard input constraint satisfaction is guaranteed, because of $H_u u_{0|k} \leq h_u$, which does not rely on sampling. On the other hand, for all $j = 1, \ldots, p$, we have $\mathbb{D} \subseteq \mathbb{X}_1^{S,j}$. Hence, by Proposition 1, for all feasible $(x_k, \mathbf{v}_k) \in \mathbb{D}$, we can ensure with confidence $(1-\delta)$ that the chance constraint (2a) is satisfied. □

Finally, we analyze the convergence properties of the proposed scheme. To this end, we first remark that, since additive disturbances affect the system at every time instant, we cannot expect the closed-loop system to be asymptotically stable at the origin. However, we can show that, under persistent noise excitation, the closed-loop state at time k+1 does remain bounded even if the candidate solution, i.e. the previously planned trajectory, may not remain feasible with given probability. First, let us formally define the candidate solution as follows.

Definition 1 (Candidate Solution)**.** *Given the OS-SMPC optimization problem in (16) and a feasible solution $v_{T|k}$ at time k, the candidate solution $\tilde{v}_{\ell|k+1}$ at time k+1 is defined as*

$$\tilde{v}_{\ell|k+1} = \begin{cases} v_{\ell+1|k} + K A_{cl}^\ell B_w w_k, & \ell = 0, \ldots, T-2 \\ K A_{cl}^\ell B_w w_k, & \ell = T-1 \end{cases}.$$

Then, under the following assumption, we can prove that the cost increase is bounded if the candidate solution does not remain feasible for a given probability.

Assumption 4 (Bounded Optimal Value Function)**.** *Let $V_T(x_k)$ be the optimal value function of the quadratic program (16), and let P_ℓ, $P_u \in \mathbb{R}^{n \times n}$, $P_\ell \succ 0$, $P_u \succ 0$, $c \in \mathbb{R}$ be such that $x_k^T P_\ell x_k \leq V_T(x_k) - c \leq x_k^T P_u x_k$ holds for all $x_k \in \mathbb{C}^\infty_{T,x}$.*

We are now in the position to state the main result of this section, i.e. the asymptotic average stage cost converges to a steady-state value, which is finite. Indeed, due to the presence of additive disturbance, the system does not asymptotically converge to the origin but it remain in its neighborhood, "oscillating" with a bounded variance, as proved in the following Proposition. The proof is reported in Appendix B.

Proposition 3 (Asymptotic Bound)**.** *Let $\varepsilon_f = [0, 1)$ be the maximum probability that the previously planned trajectory is*

not feasible. Then, there exists a constant $C = C(\varepsilon_f)$ *such that*

$$\lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t} \mathbb{E}\left\{\|x_k\|_2^2\right\} \leq C. \tag{17}$$

Remark 1. *The probability* ϵ_f *is a problem-dependent parameter related to the maximum probability that the previously planned trajectory does not remain feasible. In our approach, this parameter can only be evaluated a posteriori and depends on the application and indirectly on the chosen allowed probability of constraint violation. However, similar to the approach discussed in [12], the constraint tightening could be modified to guarantee an user-chosen bound on* ϵ_f.

The results of this section guarantee that the proposed OS-SMPC scheme enjoys important theoretical properties. These, combined with the efficiency of the scheme, which confines all costly computations in an offline step, and the generality of the considered setup, addressing both additive noise and parametric uncertainty, render the scheme suitable for efficient real-time and safety-critical applications. In the next section, we show how the scheme can be applied to control the last stage of an ARVD mission.

III. Proximity Operations Model Setup

The objective of the following section is to investigate the applicability of the OS-SMPC to achieve autonomous docking in an ARVD mission. Goal of the control, in the docking stage, is to guide an active vehicle, the chaser, towards a passive one, the target, along a specific trajectory, while satisfying security constraints.

A. The NPS-POSEYDIN Simulator

The proposed MPC controller was experimentally evaluated at the Naval Postgraduate School (NPS) *Proximity Operation with Spacecraft: Experimental hardware-In-the-loop DYNamic simulator* (POSEIDYN), an experimental testbed developed to provide a representative system-level platform upon which to develop, experimentally test, and partially validate GNC algorithms.

As shown in Figure 2, the NPS-POSEIDYN consists of four main elements: (i) a 15 ton, 4-by-4 meter polished granite monolith, with a planar accuracy of ± 0.0127 mm and a horizontal leveling accuracy at least 0.01 deg; (ii) two Floating Spacecraft Simulators (FSS), representing real spacecraft, which use three 25 mm air bearings, which use compressed air to lift the FSS approximately 5 μm, to float on top of the granite table emulating orbital spacecraft moving in close proximity of another vehicle or object (see Figure 3) thanks to eight cold-gas thrusters used to propel the FSS, each one providing a maximum thrust of 0.15 N; (iii) a commercial motion capture system, produced by British Vicon Motion Systems Ltd [33], composed by ten overhead cameras, which accurately determines the position of objects carrying passive markers (i.e. the FSS); (iv) a ground station computer, connecting the station with both the FSS and the Vicon system via Wi-Fi. The onboard computational capabilities of the FSS are provided by a PC-104 form-factor onboard computer,

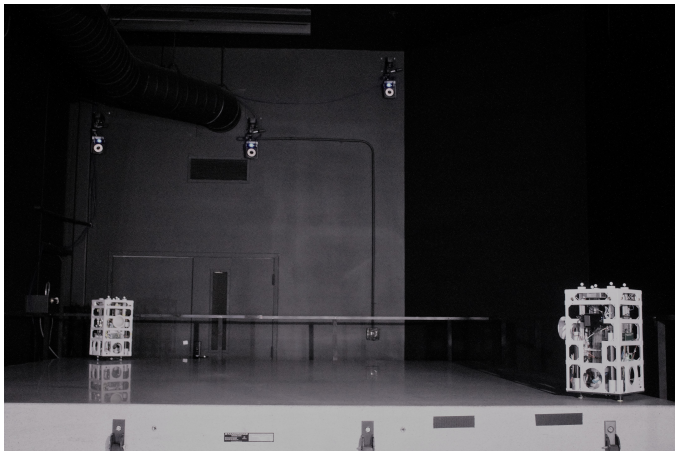


Fig. 2. NPS-POSEIDYN testbed with the Vicon motion capture cameras, FSSs, and granite monolith in the Spacecraft Robotics Laboratory at the Naval Postgraduate School. The target FSS is on the right and the chaser FSS is on the left. For obvious reasons, the applicability of the testbed, as a high-fidelity dynamic simulator, is limited to short lived close proximity operations, with respect to the planar motion only.

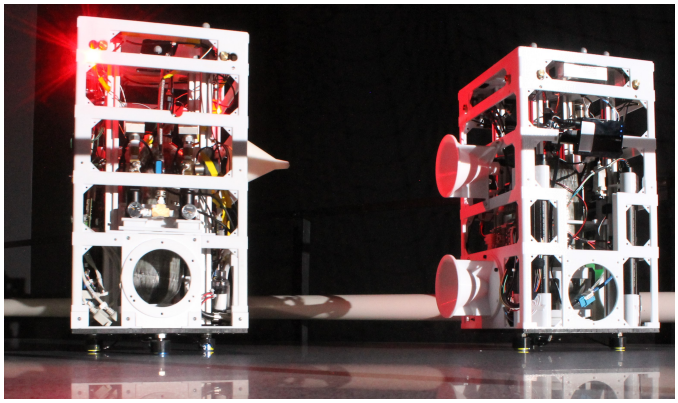


Fig. 3. NPS-POSEIDYN FSSs: the chaser on the left and the target on the right.

based on an Intel Atom 1.6 GHz 32-bit processor, with 2 GB of RAM and an 8 GB solid-state drive. Moreover, as described in [34], a real-time operating system (OS) represents the core of the FSS software architecture and the desired real-time requirement is ensured by the adoption of a Ubuntu 10.04, 32-bit server-edition OS and its Linux kernel 2.6.33. The multi-rate GNC software running atop the RT-Linux OS is developed utilizing MATLAB/Simulink environment, and once developed, the Simulink models are autocoded to C, compiled and sent from the ground station to the FSS via Wi-Fi, loading the software on the FSS on-board computer.

The main FSS physical properties are resumed in Table I, in terms of mass, geometry, and moment of inertia (MOI). Additional information about the testbed are provided in [34], where the simplified procedure to develop and consequently implement the GNC algorithm onboard the FSS are exhaustively described.

TABLE I
SUMMARY OF RELEVANT FSS PHYSICAL PROPERTIES.

Parameter	Value
Dry Mass [kg]	9.465 ± 0.001
Wet Mass [kg]	9.882 ± 0.001
Dimensions [m]	$0.27 \times 0.27 \times 0.52$
MOI [kg·m^2]	0.2527

B. Model of the Planar Experimental Testbed

To design the control architecture, we started by deriving a continuous-time description of the Chaser dynamics, taking into account parametric uncertainties and additive noise, obtaining an uncertain state-space equation of the form

$$\dot{x} = A(q)x + B(q)u + B_w w, \tag{18}$$

in which w is the vector of additive disturbance and q is the vector of parametric uncertainty, defined according to Assumption 1 and Assumption 2, respectively. In our setup, the additive noise term, which is modeled as a random and bounded model (truncated Gaussian), is related to the external environment, in which the experimental tests will be performed. The uncertainties in the state-space model are due to several sources: (i) discrepancies between the mathematical model and the actual dynamics of the physical system in operation, as linearization effects and neglected high-order dynamics; (ii) parametric physical uncertainties, such as mass and MOI variation due to fuel consumption, characterized by a uniform distribution.

In particular, we describe next how linearization introduces important uncertainty sources in the state-space model. The linearized relative dynamics of the chaser with respect to the target vehicle during the final approach of the rendezvous maneuver, modeled as two double integrators, has been derived by Clohessy and Wiltshire in [35], starting from the nonlinear equations for the restricted three-body problem and considering for both the spacecraft on a reference circular orbit around a master body. Considering the two spacecraft masses infinitesimal with respect to the mass of the main body (reference planet), we define $\rho = \rho \mathrm{i}_\rho$ and $\mathrm{r}_1 = r_1\,\mathrm{i}_\xi$ as the position vectors of the chaser and the target spacecraft respectively, where i_ρ and i_ξ represent the unit vector in the direction of the main body-chaser and the main body-target, respectively. Then, letting $\mathrm{r} = r\,\mathrm{i}_\xi$ the vectorial sum of the two positions, $\mathrm{r} = \rho + \mathrm{r}_1$, the equations of motion of the chaser spacecraft can be rewritten as

$$\frac{d^2\rho}{dt^2} + 2\omega \times \frac{d\rho}{dt} + \omega \times [\omega \times (\rho + \mathrm{r}_1)] = -\frac{\omega^2 r_1^3}{r^3}\mathrm{r}, \tag{19}$$

where ω is the orbital angular rate. Note that this differential equation presents nonlinearities due to the term $1/r^3$. In [35], using a Taylor Series expansion, a linear equation was obtained by ignoring the high order terms $\mathrm{O}(\rho^2/r_1^2)$, as $\frac{r_1^3}{r^3} = 1 - 3\,\mathrm{i}_\xi \cdot \mathrm{i}_\rho \frac{\rho}{r_1} + \mathrm{O}(\frac{\rho^2}{r_1^2})$. That is, Eq. (19) reduces to the linearized differential equation for the motion of the chaser relative to the target spacecraft as

$$\frac{d^2\rho}{dt^2} + 2\omega \times \frac{d\rho}{dt} = -\omega^2\zeta\mathrm{i}_\zeta + 3\omega^2\xi\mathrm{i}_\xi + \mathrm{O}(\rho^2). \tag{20}$$

Ignoring the $\mathrm{O}(\rho^2)$ and expressing the position vector in a more convenient way as

$$\rho \equiv \mathrm{r} = x\,\mathrm{i}_\theta + z\,\mathrm{i}_r - y\,\mathrm{i}_y, \;\; \mathrm{i}_{r_1} = \mathrm{i}_r \;\; \omega = -\omega\,\mathrm{i}_y, \tag{21}$$

with x in the direction of the motion i_θ, z in the radial direction i_r and $\mathrm{i}_y = \mathrm{i}_\theta \times \mathrm{i}_r$ normal to the orbital plane, the scalar form of the well-known Clohessy-Wiltshire (CW) Equation can be obtained. Hence, the parametric uncertainty introduced in the model are of the same order of $\mathrm{O}(\rho^2/r_1^2)$ and $\mathrm{O}(\rho^2)$. When external forces are acting on the system with a mass m_{CV}, in this case due to the correction actions actuated by the thrusters (F_x, F_y, F_z) of the Attitude and Orbit Control Subsystem (AOCS), we have

$$\begin{aligned} \frac{d^2x}{dt^2} - 2\omega\frac{dz}{dt} &= \frac{F_x}{m_{CV}}, \\ \frac{d^2y}{dt^2} + \omega^2 y &= \frac{F_y}{m_{CV}}, \\ \frac{d^2z}{dt^2} + 2\omega\frac{dx}{dt} - 3\omega^2 z &= \frac{F_z}{m_{CV}}. \end{aligned} \tag{22}$$

Considering only the in-plane motion, here defined by the x-z plane, and neglecting the terms $(-2\omega\dot{z})$, $(+2\omega\dot{x} - 3\omega^2 z)$, we get double integrators for the translational dynamics

$$\ddot{x} = \frac{F_x}{m_{CV}} \quad \ddot{z} = \frac{F_z}{m_{CV}}. \tag{23}$$

Furthermore, a double integrator is also considered for the rotational dynamics as $\ddot{\theta} = \tau/I_z$, where $\ddot{\theta}$ is the angular acceleration, τ is the control torque and I_z denotes the MOI about the vertical axis of the chaser FSS. Then, starting from the definition of the FSS dynamic model, and defining the state vector as $x = \left[\mathrm{x}, \mathrm{y}, \dot{\mathrm{x}}, \dot{\mathrm{y}}\right]^T$ and the contol vector $u = \left[F_x, F_y\right]^T$, a continuous-time linearized model of the form (18). Then, after discretization, we obtained the following discrete-time representation of the FSS uncertain dynamics as

$$x_{k+1} = A(q_k)x_k + B(q_k)u_k + B_w w_k \tag{24}$$

where $x_k \in \mathbb{R}^4$ is the state vector at time k, $u_k \in \mathbb{R}^2$ is the control input, and $w_k \in \mathbb{R}^4$ and $q_k = [q_1, q_2, q_3, q_4] \in \mathbb{R}^4$ are the vectors of the additive disturbance and the parametric uncertainty, respectively. In particular, the uncertainty vector q_k takes into account the linearization errors previously discussed, and the parametric uncertainty due to the mass variation. The corresponding continuous uncertain state and control matrices are

$$A(q) = \begin{bmatrix} q_1 & 0 & 1 & 0 \\ 0 & q_1 & 0 & 1 \\ 0 & 2q_2 & 0 & 0 \\ 0 & 3q_3 & -2q_2 & 0 \end{bmatrix}, B(q) = \begin{bmatrix} & 0_{2\times 2} & \\ \frac{1}{m} + q_4 & & 0 \\ 0 & & \frac{1}{m} + q_4 \end{bmatrix}. \tag{25}$$

All the described uncertainty sources were taken into account in constructing the linearized state and control matrices defined in (25). In particular, the parametric uncertainties

q_1, q_2, q_3 take into account linearization effects and multiplicative disturbances, and are approximated as iid random variables with uniform distribution: $q_1 \sim \mathcal{U}[5 \cdot 10^{-5}, 5 \cdot 10^{-4}]$, $q_2 \sim \mathcal{U}[0.001, 0.0014]$, $q_3 \sim \mathcal{U}[10^{-6}, 1.44 \cdot 10^{-6}]$, while q_4 refers to uncertainty in the mass, and is expressed as $q_4 \sim \mathcal{U}[-0.0091, 10^{-4}]$. Furthermore, the system is affected by persistent bounded disturbances $w \in \mathbb{R}^4$, described as a truncated Gaussian with zero mean value and unitary covariance, bounded in the set $\mathbb{W} \doteq \left\{ w \in \mathbb{R}^4 \,|\, \|w\|_\infty \leq 5 \cdot 10^{-3} \right\}$.

The focus of this experimental campaign is to investigate the performance of the OS-SMPC algorithm in the control of the translational dynamics of the chaser during the last part of the rendezvous maneuver. Attitude control of the FSS was achieved through a Tube-based Robust MPC (TRMPC) approach, already experimentally validated in [31]. The requirement of (deterministic) robust control for the attitude was driven by the physical characteristic of the docking mechanisms, located on both the FSS. Indeed, docking is ensured by an attractive force generated by the magnets on the docking interfaces, which requires a fine alignment of the two vehicles. The TRMPC was hence adopted to align and maintain the FSS pointing at the desired attitude, with respect to the target one.

Goal of the translational control is to drive the chaser to the docking position, where the target is located, while guaranteeing the satisfaction of the typical position and velocity constraints applied to the proximity maneuver. It is important to clarify that the x-y coordinate system of the testbed "coincides" with the x-z orbital plane of (23). In particular, the trajectories should lie in a desired *approach cone* (see Figure 4), i.e. LOS-like constraint, whose polytope vertices are defined as follows: $\chi_1 = (0, 0), \chi_2 = (4, 2.25), \chi_3 = (2.25, 4)$. The target is located in the suitable terminal region, determined according to Assumption 3. From the state constraint polytopes, linear inequality constraints can be derived. Additionally, the approaching and terminal velocities are bounded according to soft docking constraints. These constraints on the state are expressed in terms of chance constraints of the form (2a).

Moreover, the thrusters of the chaser are limited by a saturation constraint, according to the maximum thrust available for each cold thruster equipped on the FSS. This is an hard input constraints of the form (2b), that is

$$u_k \in \mathbb{U} = \left\{ u \in \mathbb{R}^2 \,|\, \|u\|_\infty \leq 0.3 \right\},$$

since at most two thrusters can be fired contemporary in the same direction.

C. Real-time Implementability

In this section, we discuss implementation issues related to real-time applicability of the proposed scheme, showing how it is possible to envisage the application of an OS-SMPC in an onboard implementation. In fact, this is due to the offline uncertainty approach, which significantly lowers the online computational effort with respect to other sampling-based method which require online sampling, as e.g. [20], [21]. Anyway, it should be remarked the the computational cost of the proposed OS-SMPC approach is negatively affected by the possibly high number of constraints involved in the optimization problem definition. Hence, a meticulous analysis of the solver to be implemented in the embedded microcontroller is still mandatory, together with the simplification of the constraint polytope.

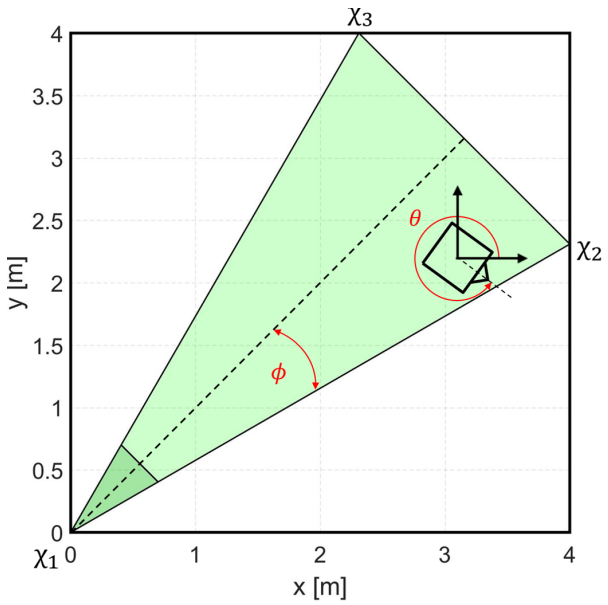


Fig. 4. NPS-POSEIDYN testbed with the cone constraints. The chaser initial condition has to be chosen within the feasible region (light green) whereas the target spacecraft can be located within the feasible terminal region (dark green). ϕ defines the cone half-angle, whereas θ represents the chaser FSS attitude with respect to the testbed reference system.

In this regard, it should be pointed out that the OS-SMPC proposed in this work has before never been implemented in real-time applications, and more generally the validation in realistic simulation environments of scenario programs as well as sampling-based SMPC approaches [12], [5] is rather limited. For this reason, a deep analysis of the available solvers has been performed to find the best one able to deal with a very high number of constraints and compliant with online implementation and low computational power hardware. Several solvers have been tested to evaluate their computational capabilities and limitations with respect to embedded implementation. Moreover, since hardware GNC software running on the FSS is developed in a MATLAB/Simulink environment, the selection criteria for the solver analyzed was driven by the compatibility with this environment and available MATLAB interface. The tested solvers were: (i) IBM ILOG CPLEX Optimizer [36], a decision optimization software developed by IBM which provides flexible, high-performance mathematical programming solvers also for quadratic programming problems; (ii) Mosek [37], a tool for solving mathematical optimization problems such as convex quadratic problems based on a powerful state-of-the-art interior-point optimizer; (iii) Gurobi Optimizer [38], a state-of-the-art solver for mathematical programming, designed from the ground up to exploit modern architectures and multi-core processors, using the most advanced implementations of the latest algorithms,

including a quadratic programming solver; (iv) *quadprog*, the interior-point-convex algorithm provided by the MATLAB Optimization Toolbox to solve quadratic programming problem; (v) *fastmpc* exploits the structure of the quadratic programming that arise in MPC, obtaining an innovative online optimization tool, based on an interior-point method, able to evaluate the control action about 100 times than a method that uses a generic optimizer, as presented in [39]; (vi) *quadwright*, a quadratic programming solver developed by J. Currie at al., presented in [40], able to speed up the computational capabilities for embedded applications.

IBM CPLEX and Gurobi are commercial softwares that provide quite easy MATLAB interfaces, enabling the user access to higher performing state-of-the-art solvers. However, both optimizers are not hardware-driven even if they provide embedding methods, and they showed bad memory leaks when calling the solver many times. Mosek is a tool for solving mathematical optimization problems, and in particular, convex quadratic problems. The software provides replacements for some MATLAB functions, including *quadprog*, and showed a rather high computational time when facing the large number of constraints involved in our setup. The MATLAB *quadprog* gives the possibility to choose between two different approaches: (i) an interior-point-convex method; and (ii) an active-set method. The first algorithm handles only convex problems whereas the second one, identified as trust-region-reflective algorithm, is able to manage problems with only bounds, or only linear equality constraints, but not both. In both cases, MATLAB *quadprog* showed slower performance than Mosek, and moreover it cannot be C-compiled. For what concerns the *fastmpc* solver, it has been developed to speed up MPC computational time and it has been proved to be able to compute in approximately 5ms the control actions for a problem with 12 states, 3 inputs, 30 as prediction horizon and about 1300 constraints. However, even if the number of states and inputs was lower for our problem, as well the prediction horizon is smaller, the much higher number of constraints resulted in a degeneration of its performance.

Our final choice fell on the quadratic programming solver *quadwright*. This very fast solver, developed with a focus on efficient memory use, ease of implementation, and high speed convergence, is based on the optimization algorithm proposed in [41]. This approach has been specifically developed to handle the core problem in MPC, namely control of a linear process with quadratic objectives subject to general linear inequality constraints. In particular, the algorithm does not exploit sparsity and it has been refined by pre-factorizing where possible, using the Cholesky Decomposition factorization when required, and heuristic for warm start, as reported in [40].

IV. Simulation and Experimental Results

In this section, we present both simulation and experimental results related to the application of the OS-SMPC scheme to control the uncertain FSS system dynamics in the last-part of the ARVD phase. To this end, we first set the probabilistic parameters of the state chance constraints as $\varepsilon_\alpha = \varepsilon_\beta = \varepsilon_\gamma = 0.05$ and $\delta = 10^{-3}$ (they should be satisfied with probability of 95% and confidence 99.9%). The ensuing number of samples $N_{tot} = N^x + N^u + N^T$ is equal to $32,370$, following the approach of [5] for $\varepsilon \in (0, 0.14)$. Then, MPC cost weight matrices were set to $Q = \text{diag}\left\{10^4, 10^4, 10^8, 10^8\right\}$ and $R = \text{diag}\left\{10^6, 10^6\right\}$, and the prediction horizon to $T = 10$. An appropriately robustly stabilizing feedback gain matrix K was designed offline using classical robust tools [42].

The main sample times set for the FSS model are reported in Table II. The initialization settings introduced here have been adopted both for simulations and experiments, to be as conservative as possible and obtain comparable results. In particular, the sample time for OS-SMPC has been set in compliance with the real-time implementability for the experimental validation.

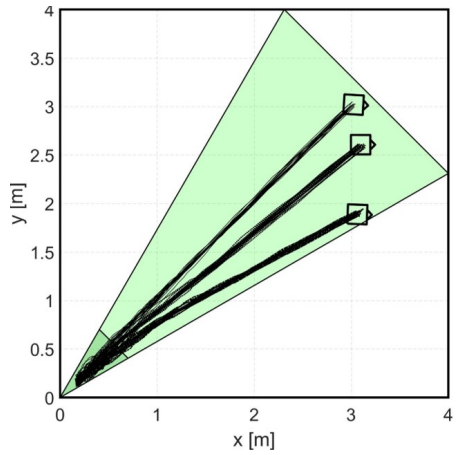
TABLE II
Model initialization settings.

Parameter	Sample Time [s]
Sensors, Actuators, Telemetry	0.01
Navigation	0.02
TRMPC, SMPC	5

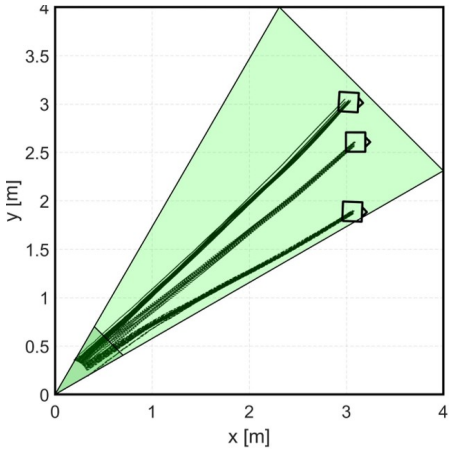
Samples of the uncertainty and of the noise sequence were extracted offline and the constraint sets (9), (11), and (12) were derived offline, leading to a total of 956,752 linear inequality constraints. Then, an iterative reduction procedure was applied leading to a final reduced constraint set of the form (13), composed by only 10,125 constraints. Once the first step constraint (15) has been obtained and intersected with (13). This completed the offline part of the OS-SMPC scheme.

The OS-SMPC algorithm was first validated by MATLAB simulations, and subsequently applied to the NPS-POSEIDYN system. It should be remarked that preliminary simulation results were presented in [43] in which 100 trajectories, each one for a feasible random initial condition (IC), were simulated. In this paper, considering the NPS-POSEIDYN setup and the diagonal symmetry of both the granite monolith and the cone constraint, the ICs for the OS-SMPC simulated and experimental validation were set only in one half of the plane. Thus, three case studies corresponding to three relevant ICs were chosen due to their peculiarities: (i) the first IC represents the diagonal case, in which the chaser FSS is farthest from the cone boundaries (case A); (ii) the second IC is the most critical IC, since the FSS is very close to cone constraint (case B); (iii) the last case represents the halfway condition (case C).

Each case study was simulated and subsequently experimentally reproduced several times, to validate the behavior of the controller. The results obtained are represented in Figure 5, which depict 20 repetitions for each IC, both for simulations and experiments. Comparing the simulation trajectories (Figure 5(a)) with the experimental ones (Figure 5(b)), we

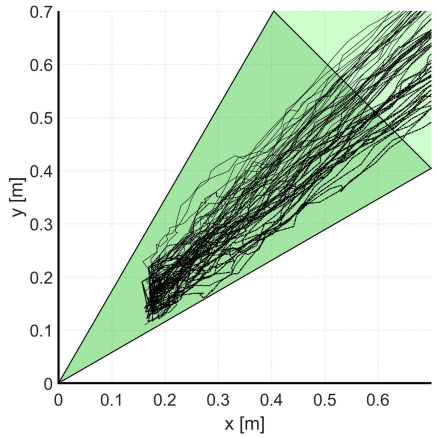


(a) Simulation Results

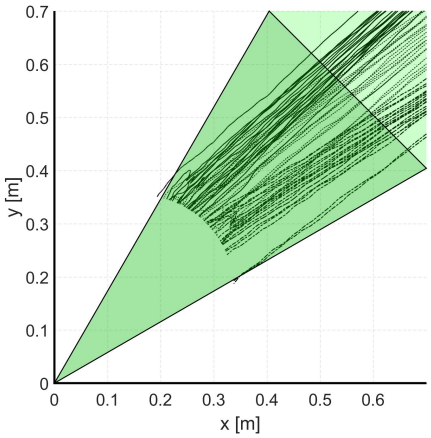


(b) Experimental Results

Fig. 5. Simulation and experimental results for 3 different ICs, considering 20 repetitions for each one.



(a) Simulation Results



(b) Experimental Results

Fig. 6. Zoom on the terminal region of both simulation and experimental results.

observed a rather good adherence of the results. In particular, in all experiments the chaser FSS is always driven from the IC to the terminal region, where the target FSS is located.

A zoom-in of the terminal region, both for simulations and experiments, is reported in Figure 6. We notice a relevant difference between Figure 6(a) and Figure 6(b) with respect to the stopping condition. In the simulations, the chaser stops when the relative distance with respect to the *virtual* target Center of Mass (CoM) is lower than a certain threshold (0.18 m). On the other hand, in the experimental setup, the target is a real FSS, which is equipped with a *female* magnetic docking mechanism. Similarly, the chaser FSS has a *male* interface. Hence, the end of the docking phase between the spacecraft is due to the magnetic force generated between the two magnets. The effects of this force are evident in Figure 6(b), where trajectories are not funneled as in Figure 6(a) but they are distributed around the target docking interface. This discrepancy is mainly due to the fact that the magnetic force was not introduced in simulation. Still, the probability of constraint violation has been kept below 5% as expected, i.e. only one time over 30 the FSS went outside the constraint set while completing the maneuver.

Once the OS-SMPC scheme has been validated for the real-time implementability point-of-view, the results were analyzed also with respect to the following performance parameters:

1) Time-to-dock t_{td}, defining as the time required to the chaser FSS to reach and dock the target one, starting from the initial condition;
2) Control effort f_c, an estimate of the fuel consumption required for the maneuver, which represents also the efficiency of the control approach from an application point-of-view. The control effort can be evaluated as

$$f_c = \sum_{k=0}^{t_{td}} \|u_k\|_1 \Delta t, \tag{26}$$

where Δt represents the system sample time.

Figure 7 depicts the control effort for all 60 experiments as a function of the time-to-dock. As we can notice, in all three cases the maneuver lasted about $120 - 200$ s, with an average control effort between 4 Ns and 5 Ns.

In order to assess the effectiveness of the proposed OS-SMPC approach, the average control effort for all the experiments can be compared with the control effort obtained applying other two MPC approaches validated for the same

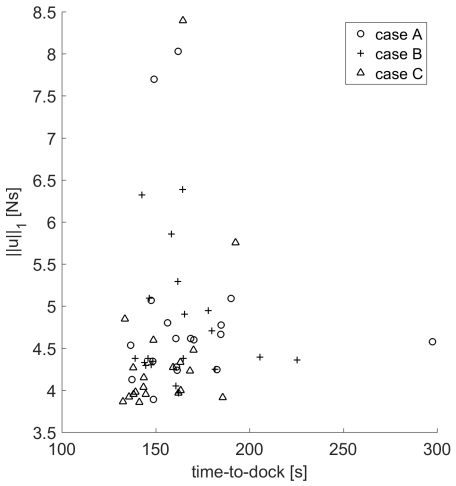


Fig. 7. Control effort with respect to the time-to-dock results for 60 experiments.

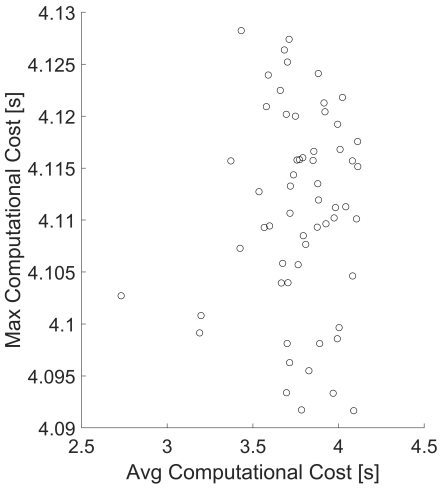


Fig. 8. Average and maximum computational cost of all 60 experiments.

maneuver and using the same testbed: a Linear-Quadratic MPC (LQMPC) and a TRMPC. In particular, the performance of the Tube-based robust MPC controller has been evaluated and compared with a classical LQMPC scheme, both in simulations and on the same experimental setup (see [44] for further details).

TABLE III
COMPARISON OF THE AVERAGE CONTROL EFFORT FOR THREE DIFFERENT MPC APPROACHES ADOPTED TO CONTROL THE FSS DURING A RENDEZVOUS MANEUVERS ON THE NPS-POSEIDYN TESTBED: (I) LQMPC; (II) TRMPC; (III) OS-SMPC.

MPC approach	Control Effort [Ns]
LQMPC	4.99
TRMPC	14.24
OS-SMPC	4.69

In Table III, the average control effort of the three MPC approaches are reported. We notice that the robust MPC scheme represents the most fuel-consuming approach, with a fuel demand about three times higher than the classical and stochastic MPC, which instead are characterized by comparable fuel consumption, in the order of 5 Ns. The fact that OS-SMPC has much lower fuel consumption than not only TRMPC but also LQMPC is somehow surprising, but it can be explained by the much lower conservatisms of the stochastic approach.

For completeness, the computational cost of all 60 experiments is provided in Figure 8 in terms of average and maximum cost and it is possible to notice that all of them are close to 4s, which leaves 1s to be computationally dedicated to the other processes, considering the MPC sample time equal to 5s.

V. CONCLUSIONS

An offline sampling-based Stochastic Model Predictive Control (OS-SMPC) algorithm is proposed for discrete-time linear systems subject to both parametric uncertainties and additive disturbances, and its theoretical properties are assessed. Real-time implementability of guidance and control strategies for automated rendezvous and proximity operations between spacecraft is proven and validated on an experimental testbed. Parametric uncertainties due to the mass variations during operations, linearization errors, and disturbances due to external space environment are simultaneously considered. The offline sampling approach in the control design phase is shown to reduce the computational cost, which usually constitutes the main limit for the adoption of SMPC schemes, especially for low-cost on-board hardware, and to provide a very effective control in terms of time-to-dock and fuel consumption. These characteristics are demonstrated both through simulations and by means of experimental results.

APPENDIX A
QUADRATIC COST MATRIX DEFINITION

Simple algebraic manipulations show that the terms in (6) can be written as follows

$$\Phi^0_{\ell|k}(\mathbf{q}_k) = A^{cl}(q_{\ell-1|k})A^{cl}(q_{\ell-2|k})\cdots A^{cl}(q_{0|k}),$$

$$\Phi^v_{\ell|k}(\mathbf{q}_k) = \begin{bmatrix} A^{cl}(q_{\ell-1|k})\cdots A^{cl}(q_{1|k})B(q_{0|k}) \\ \vdots \\ B(q_{\ell-1|k}) \\ 0_{n\times(T-\ell)m} \end{bmatrix}^T,$$

$$\Phi^w_{\ell|k}(\mathbf{q}_k) = \begin{bmatrix} A^{cl}(q_{\ell-1|k})\cdots A^{cl}(q_{1|k})B_w(q_{0|k}) \\ \vdots \\ B_w(q_{\ell-1|k}) \\ 0_{n\times(T-\ell)m_w} \end{bmatrix}^T,$$

$$\Gamma_\ell = [0_{m\times\ell m} \;\; I_m \;\; 0_{m\times(T-\ell-1)m}].$$

Then, defining the matrix

$$\Phi_T(\mathbf{q}_k) \doteq \begin{bmatrix} \Phi^0_{0|k}(\mathbf{q}_k) & \Phi^v_{0|k}(\mathbf{q}_k) & \Phi^w_{0|k}(\mathbf{q}_k) \\ \vdots & \vdots & \vdots \\ \Phi^0_{T|k}(\mathbf{q}_k) & \Phi^v_{T|k}(\mathbf{q}_k) & \Phi^w_{T|k}(\mathbf{q}_k) \end{bmatrix},$$

and considering $\bar{Q} = I_T \otimes Q$, $\bar{R} = I_T \otimes R$, $\bar{K} = I_T \otimes K$, and defining $\Gamma = [0_{mT\times n}\ \ I_{mT}\ \ 0_{mT\times m_w T}]$, the two terms, Q_E and R_E of the explicit cost matrix $\tilde{S}$

$$\tilde{S} = \mathbb{E}\left\{(Q_E + R_E)\right\}, \tag{27}$$

can be written as:

$$Q_E = M^T \Phi_T^T(\mathbf{q}_k) \begin{bmatrix} \bar{Q} & 0_{nT\times n} \\ 0_{n\times nT} & P \end{bmatrix} \Phi_T(\mathbf{q}_k) M$$

$$R_E = M^T [\bar{K}\Phi_T(\mathbf{q}_k) + \Gamma]^T \bar{R} [\bar{K}\Phi_T(\mathbf{q}_k) + \Gamma] M$$

where the matrix M is

$$M = \begin{bmatrix} \mathbb{I}_n & 0_{n\times mT} & 0_{n\times nT} \\ 0_{mT\times n} & \mathbb{I}_{mT} & 0_{mT\times nT} \\ 0_{m_w T\times n} & 0_{m_w T\times mT} & \mathbf{w}\mathbb{I}_{m_w T} \end{bmatrix}$$

As discussed in [5], matrix $\tilde{S}$ may be approximated via random sampling by exploiting classical Monte Carlo or Quasi Monte Carlo tools, as those presented in e.g. [45].

Appendix B
Proof to Asymptotic Bound

If the candidate solution does not remain feasible, the cost increase can be bounded through the matrices in Assumption 4. Let $V_T(x_k) = J_T(x_k, \mathbf{v}_k^*)$ be the optimal value of (16) at time k and consider the optimal value function of the online optimization program as stochastic Lyapunov function. Hence, if the candidate solution $\tilde{\mathbf{v}}$ remains feasible, we have

$$\begin{aligned}
&\mathbb{E}\left\{V_T(x_{k+1}) \mid x_k,\ \tilde{\mathbf{v}}_{k+1} \text{ feasible}\right\} - V_T(x_k) \\
&\le \mathbb{E}\left\{J_T(x_{k+1}, \tilde{\mathbf{v}}_{k+1}) \mid x_k\ \right\} - V_T(x_k) \\
&\le \mathbb{E}\left\{\sum_{l=0}^{T-1}(\|\tilde{x}_{\ell|k+1}\|^2_Q + \|\tilde{u}_{\ell|k+1}\|^2_R) + \|\tilde{x}_{T|k+1}\|^2_P\right\} \\
&- \mathbb{E}\left\{\sum_{l=0}^{T-1}(\|x^*_{\ell|k}\|^2_Q + \|u^*_{\ell|k}\|^2_R) + \|x^*_{T|k}\|^2_P\right\} \\
&= \mathbb{E}\Big\{\|x^*_{T|k}\|^2_{Q+K^TRK-P} + \|A_{cl}(q_k)x^*_{T|k} + B_w(q_k)w_{T|k}\|^2_P \\
&- \ \|x^*_{0|k}\|^2_Q - \|u^*_{0|k}\|^2_R\Big\} \\
&\le \mathbb{E}\Big\{\|x^*_{T|k}\|^2_{Q+K^TRK-P} + \|A_{cl}(q_k)x^*_{T|k}\|^2_P \\
&+ \ \|B_w(q_k)w_{T|k}\|^2_P + 2(A_{cl}(q_k)x^*_{T|k})^T P(B_w(q_k)w_{T|k}) \\
&- \|x^*_{0|k}\|^2_Q - \|u^*_{0|k}\|^2_R\ \}.
\end{aligned}$$

According to the definition of the terminal set (Assumption 3), we obtain

$$\begin{aligned}
&\mathbb{E}\Big\{\|x^*_{T|k}\|^2_{Q+K^TRK-P+A_{cl}(q_k)^TPA_{cl}(q_k)} \\
&+ \ \|B_w(q_k)w_{T|k}\|^2_P - \|x^*_{0|k}\|^2_Q - \|u^*_{0|k}\|^2_R\Big\} \\
&\le \mathbb{E}\Big\{\|B_w(q_k)w_{T|k}\|^2_P - \|x^*_{0|k}\|^2_Q - \|u^*_{0|k}\|^2_R\Big\} \\
&\le \mathbb{E}\left\{\|B_w(q_k)w_{T|k}\|^2_P\right\} - \|x_k\|^2_Q - \|u_k\|^2_R.
\end{aligned}$$

On the other hand, if the candidate solution is not feasible, we get

$$\begin{aligned}
&\mathbb{E}\left\{V_T(x_{k+1}) \mid x_k, \tilde{\mathbf{v}}_{T|k+1} \text{ not feasible}\right\} - V_T(x_k) \\
&\le \max_{\substack{w\in\mathbb{W} \\ (A(q_k),B(q_k))\in\mathbb{G}}} \|A(q_k)x_k + B(q_k)u_k + B_w(q_k)w_k\|^2_{P_u} - \|x_k\|^2_{P_\ell} \\
&\le \max_{\substack{w\in\mathbb{W} \\ (A(q_k),B(q_k))\in\mathbb{G}}} \Big(\|A_{cl}(q_k)x_k + B(q_k)v_k\|^2_{P_u} + \|B_w(q_k)w_k\|^2_{P_u} \\
&+ 2\|(A_{cl}(q_k)x_k + B(q_k)v_k)^T P_u B_w(q_k)w_k)\|\Big) - \|x_k\|^2_{P_\ell}.
\end{aligned}$$

Applying Cauchy-Schwarz Inequality first, and then Young Inequality, we have

$$\begin{aligned}
&\max_{\substack{w\in\mathbb{W} \\ (A(q_k),B(q_k))\in\mathbb{G}}} \Big(\|A_{cl}(q_k)x_k + B(q_k)v_k\|^2_{P_u} + \|B_w(q_k)w_k\|^2_{P_u} \\
&+ 2\|(P_u^{1/2}(A_{cl}(q_k)x_k + B(q_k)v_k))^T(P_u^{1/2}B_w(q_k)w_k)\|\Big) \\
&- \|x_k\|^2_{P_\ell} \\
&\le \max_{\substack{w\in\mathbb{W} \\ (A(q_k),B(q_k))\in\mathbb{G}}} \Big(2\|A_{cl}(q_k)x_k + B(q_k)v_k\|^2_{P_u} \\
&+ 2\|B_w(q_k)w_k\|^2_{P_u}\Big) - \|x_k\|^2_{P_\ell} \\
&\le 2 \max_{(A(q_k),B(q_k))\in\mathbb{G}} \Big(\|A_{cl}(q_k)x_k + B(q_k)v_k\|^2_{P_u} \\
&+ 2\max_{w\in\mathbb{W}} \|B_w(q_k)w_k\|^2_{P_u} - \|x_k\|^2_{P_\ell}\Big).
\end{aligned}$$

Let $\lambda_{min}(q_k)$ be a lower bound on the smallest eigenvalue of

$$U(q_k) = \begin{bmatrix} Q - \frac{2\varepsilon_f}{1-\varepsilon_f}(A(q_k)^T P_u A(q_k) - \frac{1}{2}P_\ell) & -\frac{2\varepsilon_f}{1-\varepsilon_f}A(q_k)^T P_u B(q_k) \\ -\frac{2\varepsilon_f}{1-\varepsilon_f}B(q_k)^T P_u A(q_k) & R - \frac{2\varepsilon_f}{1-\varepsilon_f}B(q_k)^T P_u B(q_k) \end{bmatrix}, \tag{33}$$

that is $\lambda_{min} \le \min_{q_k\in\mathbb{Q}}\ (\ \min_{i=1,\dots,n+m} \lambda_i(U(q_k)))$. Hence, applying the law of total probability

$$\begin{aligned}
&\mathbb{E}\left\{V_T(x_{k+1}) \mid x_x, \tilde{\mathbf{v}}_{T|k+1}\right\} - V(x_k) \\
&\le (1-\varepsilon_f)\Big(\mathbb{E}\left\{\|B_w(q_k)w^*_{T|k}\|^2_P\right\} - \|x_k\|^2_Q - \|u_k\|^2_R\Big) \\
&+ \varepsilon_f\Big(2 \max_{(A(q_k),B(q_k))\in\mathbb{G}} \|A_{cl}(q_k)x_k + B(q_k)v_k\|^2_{P_u} \\
&+ 2\max_{w\in\mathbb{W}} \|B_w(q_k)w_k\|^2_{P_u} - \|x_k\|^2_{P_\ell}\Big) \\
&\le -(1-\varepsilon_f)\lambda_{min}\|x_k\|^2_2 + (1-\varepsilon_f)\mathbb{E}\left\{\|B_w(q_k)w^*_{T|k}\|^2_P\right\} \\
&+ 2\varepsilon_f \max_{w\in\mathbb{W}} \|B_w(q_k)w_k\|^2_{P_u} \\
&\le -(1-\varepsilon_f)\lambda_{min}\|x_k\|^2_2 + C.
\end{aligned}$$

The final statement follows taking iterated expectations and noting that $V(x_k) < \infty$. □

REFERENCES

[1] D. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[2] E. Dreyfus, "The art and theory of dynamic programming," Tech. Rep., 1977.

[3] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and Design*. Nob Hill Pub., 2009.

[4] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust, and Stochastic*. Springer, 2015.

[5] 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.

[6] J. Grosso, C. Ocampo-Martínez, V. Puig, and B. Joseph, "Chance-constrained model predictive control for drinking water networks," *Journal of Process Control*, vol. 24, no. 5, pp. 504–516, 2014.

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

[8] D. V. 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.

[9] 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.

[10] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, "Racing miniature cars: Enhancing performance using stochastic mpc and disturbance feedback," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 5642–5647.

[11] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

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

[13] 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.

[14] M. S. Lobo and S. Boyd, "Policies for simultaneous estimation and optimization," in *Proc. of American Control Conference*, 1999.

[15] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1468–1480, 2012.

[16] L. Blackmore, M. Ono, A. Bektassov, and B. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[17] A. Mesbah, S. Streif, R. Findeisen, and R. Braatz, "Stochastic nonlinear model predictive control with probabilistic constraints," in *Proc. of American Control Conference*, 2014.

[18] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems: with Applications*. Springer Science & Business Media, 2012.

[19] 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.

[20] G. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 219–224, 2012.

[21] ——, "Stochastic model predictive control of LPV systems via scenario optimization," *Automatica*, vol. 49, no. 6, pp. 1861–1866, 2013.

[22] 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.

[23] G. C. Calafiore, "On the expected probability of constraint violation in sampled convex programs," *Journal of Optimization Theory and Applications*, vol. 143, no. 2, pp. 405–412, 2009.

[24] G. Schildbach, L. Fagiano, and M. Morari, "Randomized solutions to convex programs with multiple chance constraints," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2479–2501, 2013.

[25] National Research Council and others, *NASA Space Technology Roadmaps and Priorities: Restoring NASA's Technological Edge and Paving the Way for a New Era in Space*. National Academies Press, 2012.

[26] D. Zimpfer, P. Kachmar, and S. Tuohy, "Autonomous rendezvous, capture and in-space assembly: past, present and future," in *Proc. of 1st Space exploration conference: continuing the voyage of discovery*, 2005.

[27] L. Breger and J. How, "Safe trajectories for autonomous rendezvous of spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1478–1489, 2008.

[28] E. N. Hartley, "A tutorial on model predictive control for spacecraft rendezvous," in *Proc. of European Control Conference*. IEEE, 2015, pp. 1355–1361.

[29] Y. Luo, J. Zhang, and G. Tang, "Survey of orbital dynamics and control of space rendezvous," *Chinese Journal of Aeronautics*, vol. 27, no. 1, pp. 1–11, 2014.

[30] F. Gavilan, R. Vazquez, and E. F. Camacho, "Robust model predictive control for spacecraft rendezvous with online prediction of disturbance bound," in *Proc. of Aerospace Guidance, Navigation and Flight Control Systems AGNFCS'09*, 2009.

[31] M. Mammarella, E. Capello, H. Park, G. Guglieri, and M. Romano, "Spacecraft proximity operations via tube-based robust model predictive control with additive disturbances," in *Proc. of 68th International Astronautical Congress*. International Astronautical Federation, 2017.

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

[33] Vicon Motion Systems Limited, "Vicon tracker user guide," Tech. Rep., 2016.

[34] R. Zappulla, J. Virgili-Llop, C. Zagaris, H. Park, and M. Romano, "Dynamic air-bearing hardware-in-the-loop testbed to experimentally evaluate autonomous spacecraft proximity maneuvers," *Journal of Spacecraft and Rockets*, vol. 54, no. 4, pp. 1–15, 2017.

[35] W. Clohessy, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[36] CPLEX Users Manual, "IBM ILOG CPLEX Optimization Studio, Version 12, Release 6," Tech. Rep., 2015.

[37] MOSEK ApS, "MOSEK FAQ, Release 8.1.0.42," Tech. Rep., 2018.

[38] Gurobi Optimization, Inc., "GUROBI Optimizer Reference Manual, Version 7.5," Tech. Rep., 2017.

[39] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[40] J. Currie, A. Prince-Pike, and D. Wilson, "Auto-code generation for fast embedded model predictive controllers," in *Proc. of International Conference Mechatronics and Machine Vision in Practice*, 2012.

[41] S. Wright, "Applying new optimization algorithms to model predictive control," Tech. Rep., 1996.

[42] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE, 2004, pp. 284–289.

[43] M. Mammarella, E. Capello, M. Lorenzen, F. Dabbene, and F. Allgöwer, "A general sampling-based SMPC approach to spacecraft proximity operations," in *Proc. of Conference on Decision and Control*, 2017.

[44] M. Mammarella, E. Capello, H. Park, G. Guglieri, and M. Romano, "Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance," *Aerospace Science and Technology*, vol. 77, pp. 585–594, 2018.

[45] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*. Siam, 1992, vol. 63.

Martina Mammarella (M'18) received the B.S. and M.S. degrees, and is currently pursuing the Ph.D. degree, all in Aerospace Engineering from Politecnico di Torino, Italy. Her domains of expertise are flight mechanics, spacecraft control system, mission and system design. She worked and is currently involved in some Italian and European projects supported by the European Space Agency, Italian Space Agency, and Piedmont region. She is working on the development of a 6 dof simulator for rendezvous e docking maneuvers of an electric space tug. The main topic is the definition, design and real-time validation of compliant classical, robust and stochastic Model Predictive Control approaches for spacecraft proximity operations.

Matthias Lorenzen received the M.S. degree (Dipl.-Ing.) in engineering cybernetics from the University of Stuttgart, Stuttgart, Germany, in 2013 and is currently pursuing the Ph.D: degree in the Graduate School of Simulation Technology, Stuttgart Research Center for Simulation Technology, Stuttgart, Germany. During his studies he spent one year as DAAD Fellow at Harvard University, Cambridge, MA. In 2013, he joined the Institute for Systems Theory and Automatic Control as a Research and Teaching Assistant. His main research interests are stochastic and robust Model Predictive Control.

Elisa Capello (M'14) received her PhD (March 2011) from Politecnico di Torino and now she is in the Flight Mechanics group of the Mechanical and Aerospace Engineering Department of Politecnico di Torino as Assistant Professor. Her domains of expertise are flight mechanics, aircraft (manned and unmanned) control system, Matlab programming. She worked in some Italian projects supported by the Piedmont region and by the Italian Antarctica Research Project. She serves as regular reviewer for scientific journals, published by Springer, Emerald, ASCE, etc.. She is working on the assessment of load control system activation logics to react effectively to sudden or quasi-static external load source (as gust, turbulence), in the JTI Clean Sky project supported by EU research funds and on the development of a 6 dof simulator for rendezvous e docking maneuvers.

Hyeongjun Park (M'16) received the B.S. and M.S. degrees from Seoul National University, Seoul, Korea, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 2014, all in aerospace engineering. He was a Postdoctoral Research Associate of U.S. National Research Council at the Naval Postgraduate School, Monterey, CA, from 2015 to 2017. He is currently an Assistant Professor of the Department of Mechanical and Aerospace Engineering with New Mexico State University, Las Cruces, NM, USA. His research interests include real-time optimal control of constrained systems, guidance and control of spacecraft proximity operations, and autonomous aerial manipulation for interaction with the environment.

Fabrizio Dabbene (SM'09) is a Senior Researcher at the institute IEIIT of the National Research Council of Italy (CNR), where he is the coordinator of the Systems, Modeling & Control Group and elected member of the Scientific Council. He has held visiting and research positions at The University of Iowa, at Penn State University and at the Russian Academy of Sciences, Institute of Control Science, Moscow. He published more than 100 research papers and two books. His research interests include randomized and robust methods for systems and control, and modeling of environmental systems. Dr. Dabbene received the 2010 EurAgeng Outstanding Paper Award. He served as Associate Editor for Automatica (2008-2014) and the IEEE Transactions on Automatic Control (2008-2012). He has taken various responsibilities within the IEEE-CSS: he was elected member of the Board of Governors (2014-2016) and he served as Vice-President for Publications (2015-2016).

Giorgio Guglieri is a full professor in flight mechanics at Politecnico di Torino where he belongs to the steering committee for the doctoral course in aerospace engineering. He is senior member of AIAA and AHS. Author of several articles and reviewer of papers submitted to several international journals. He is involved in several research activities on fixed and rotary wing flight mechanics, flight control system design, applied and experimental aerodynamics, HMI interfaces design and tele-operations, UAV design and RPAS development and testing, optimal path planning and collision avoidance, advanced multi-task PC-based HMI, optimal control design based on meta-heuristic algorithms, simulation of spacecraft reentry vehicle dynamics, and rendezvous and docking, including GNC modeling. He is executive officer and technical manager of Mavtech and is the representative of the Aerospace District of Piedmont Region in the Council of ACARE Italia.

Marcello Romano (SM'04) received his M.S. in Aerospace Engineering in 1997 and his Ph.D. in Astronautical Engineering in 2001 both from Politecnico di Milano, Milan, Italy. He is a Full Professor in the NPS Mechanical and Aerospace Engineering Department and a member of the NPS Space Systems Academic Group. Prof. Romano is the founder and director of the Spacecraft Robotics Laboratory at NPS which hosts an experimental dynamic simulator testbed for spacecraft proximity maneuvers.

Frank Allgöwer (M'03) studied Engineering Cybernetics and Applied Mathematics in Stuttgart and at the University of California, Los Angeles (UCLA), respectively, and received his Ph.D. degree from the University of Stuttgart in Germany. He is the Director of the Institute for Systems Theory and Automatic Control and Executive Director of the Stuttgart Research Centre Systems Biology at the University of Stuttgart. His research interests include cooperative control, predictive control, and nonlinear control with application to a wide range of fields including system biology. At present Frank is President Elect of the International Federation of Automatic Control IFAC and serves as Vice-President of the German Research Foundation DFG.