

Minimal Constraint Violation Probability in Model Predictive Control for Linear Systems

Michael Fink, Tim Brüdigam, Dirk Wollherr, and Marion Leibold

Abstract—Handling uncertainty in model predictive control comes with various challenges, especially when considering state constraints under uncertainty. Most methods focus on either the conservative approach of robustly accounting for uncertainty or allowing a small probability of constraint violation. In this work, we propose a linear model predictive control approach that minimizes the probability that linear state constraints are violated in the presence of additive uncertainty. This is achieved by first determining a set of inputs that minimize the probability of constraint violation. Then, this resulting set is used to define admissible inputs for the optimal control problem. Recursive feasibility is guaranteed and input-to-state stability is proved under assumptions. Numerical results illustrate the benefits of the proposed model predictive control approach.

Index Terms—predictive control, recursive feasibility, input-to-state stability, robust model predictive control (RMPC), stochastic model predictive control (SMPC)

I. INTRODUCTION

Considering uncertainty presents a major challenge in the control of safety-critical systems. Depending on the application, uncertainty ranges from noise and disturbances to model and parameter inaccuracy. For example, the control of automated vehicles needs to account for sensor noise, disturbances such as wind, or unknown future behavior of other vehicles that are close enough to be considered for safety. Ideally, the control of such safety-critical systems realizes minimal risk in the presence of uncertainty.

A prominent method for the control of safety-critical systems is Model Predictive Control (MPC), due to its ability to consider input and state constraints to satisfy safety requirements [1]. In general MPC requires a model of the system to solve an optimal control problem in each time step.

When uncertainty is present, constraints are handled in a robust way by Robust Model Predictive Control (RMPC) [2]. Initially known bounds on the uncertainty allow for a guarantee on stability and recursive feasibility. Nevertheless, robustly accounting for uncertainty comes with issues. If the uncertainty bound was initially not estimated large enough, all guarantees are lost. If uncertainty bounds are chosen too large, potentially to account for rare worst-case events, RMPC becomes highly conservative.

Overcoming conservatism is addressed by Stochastic Model Predictive Control (SMPC) [3], [4]. In SMPC, constraints subject to uncertainty are handled as chance constraints. A chance constraint requires that the constraint is satisfied to a certain level, based on a chosen risk parameter, representing the acceptable risk. While a low-risk parameter results in a high probability of constraint violation, performance is improved, as rare uncertainty realizations are neglected. Again, multiple issues arise. Similar to RMPC, wrong initial

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 490649198. This work was supported by a fellowship within the IFI program of the German Academic Exchange Service (DAAD) and the Bavaria California Technology Center (BaCaTeC) grant 1-[2020-2].

The authors are with the Chair of Automatic Control Engineering, Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany (e-mail: {michael.fink, tim.brueDIGAM, dirk.wollherr, marion.leibold}@tum.de).

assumptions for the uncertainty cause feasibility issues. Furthermore, SMPC does not penalize the allowed constraint violation, e.g. the constraint violation probability is not considered in the cost function.

When developing MPC for safety-critical systems, a small probability of constraint violation may be tolerated. It is nevertheless fundamental to achieve the minimum probability of constraint violation. In addition, recursive feasibility and stability of the closed-loop system dynamics are essential. Also dealing with variations in the uncertainty as well as in the constraints has to be considered.

Whereas RMPC and SMPC partially consider these requirements, both methods are impractical for safety-critical systems due to the above-mentioned issues, e.g., uncertainty bounds must be known in advance or recursive feasibility is not guaranteed in every situation. These considerations resulted in the development of MPC with constraint violation probability minimization (CVPM) [5], [6].

A. Related Work

While there exist different approaches to RMPC, the main ideas are similar: robustly handling constraints, accounting for worst-case uncertainty realizations. An overview of RMPC approaches is given in [1, Ch. 3] and a summary of early works in [7]. The most prominent RMPC approaches are min-max MPC [8], considering maximal uncertainties, and tube-based MPC [9], [10], which defines a tube around the nominal state trajectory to tighten constraints appropriately. While RMPC was used to control safety-critical systems, e.g., a robotic manipulator [11] or an automated vehicle [12], [13], solutions are generally conservative, as no knowledge about the disturbance distribution is taken into account. Additionally, changing uncertainty bounds or distributions, as well as changing constraints cause a loss of properties, e.g., recursive feasibility or require intensive recomputation.

SMPC employs chance constraints, i.e., the probability of violating a constraint is bounded by a risk parameter. Chance constraints are difficult to handle in general as their evaluation requires computing multivariate integrals. Therefore, approaches to reformulating these probabilistic constraints into deterministic (potentially approximated) constraints are proposed: For Gaussian uncertainties and linear constraints, the chance constraints can be analytically reformulated [4]. Gaussian distributions are suitable to describe certain types of uncertainty, e.g., noise, while often safety-critical systems require considering other distributions. For this case, sampling-based SMPC approaches are suitable. Particle-based SMPC [14] approximates the chance constraint by ensuring that only a small number of samples leads to constraint violations. In Scenario Model Predictive Control (SCMPC) [15], based on [16], the number of required samples is computed depending on a risk parameter. Then, for all sampled scenarios, the constraints must be satisfied. A major issue with sampling-based SMPC is that guarantees on recursive feasibility are difficult to obtain. A combination of RMPC and SMPC is considered in [17], [18], where robust constraints are employed on the short-term horizon and chance constraints are used for long-term predictions. This approach improves performance compared to RMPC while still employing short-term robust constraints; however, proofs of recursive feasibility and stability are challenging.

The SMPC approaches, introduced in the previous paragraph, consider open-loop chance constraints, i.e., chance constraint satisfaction is only required in the optimal control problem for the open-loop prediction. Though, recursive feasibility can only be guaranteed if closed-loop constraint satisfaction is considered [19]. An alternative method to guarantee recursive feasibility is proposed in [20], proposing an additional constraint on the predicted state of the first prediction step, making recursive feasibility challenging in SMPC.

SMPC is mostly applied to applications where constraint violations are not critical, e.g., energy control in buildings [21], [22] or hybrid electrical vehicles [23], [24]. SMPC for safety-critical systems mostly focused on automated vehicles [25]–[28].

However, safety of SMPC is only specifically addressed in few works. In [29], failsafe trajectory planning is used to guarantee safety in case of infeasible SMPC optimal control problems. A different idea is presented in [30], where a least-intrusive trajectory is found if a collision is inevitable.

All previously described SMPC approaches are unable to provide recursive feasibility or stability guarantees once unexpected changes arise, potentially due to time-varying uncertainties or time-varying constraints. While slack variables can be introduced or alternative problems can be solved [31], [32] to regain feasibility, the optimization does not necessarily provide the safest possible solution. This issue is addressed in [5] for a collision avoidance scenario. There, the probability of violating a collision avoidance norm constraint in the first prediction step is minimized with CVPM-MPC. An extension of CVPM-MPC where the norm constraint is taken into account for more than one prediction step is presented in [6]. Unlike multi-objective MPC [33], [34], in CVPM-MPC, safety is not part of an objective trade-off, but safety is maximized first before other objectives are considered. However, the approach in [5] and [6] is designed particularly for obstacle avoidance, i.e., the approach is limited to norm constraints, and only uncertainty affecting the obstacle dynamics is considered.

In summary, previous MPC approaches only cover parts of the requirements for safety-critical systems. The major challenge, reasonably minimizing constraint violation probabilities for general linear systems, is still an open problem.

B. Contribution and Structure

In this work, we propose an MPC algorithm that minimizes the probability of constraint violation. This is achieved by guaranteeing constraint satisfaction whenever possible and ensuring minimal constraint violation probability whenever constraint satisfaction is not possible. We generalize, extend, and simplify the results of the CVPM-MPC method from [5] and [6]. In particular, we now replace norm constraints by general linear constraints.

We investigate the requirements for recursive feasibility and stability. We show that no additional assumptions are needed for recursive feasibility because there are no bounds on the maximum allowed constraint violation probability. Furthermore, we propose additional rather strong assumptions for proving input-to-state stability (ISS). Finally, the practical consequences of relaxing these assumption are discussed.

The remaining parts are structured as follows. Section II introduces the problem. The CVPM-MPC method for linear systems and constraints is derived in Section III, followed by details on the properties in Section IV. A numerical example is shown in Section V, demonstrating the benefits of applying CVPM-MPC to safety-critical systems. A discussion and conclusive remarks are given in Section VI and Section VII, respectively.

C. Notation

Norms are denoted by $\|\cdot\|$. We define $\|\mathbf{a}\|_{\mathbf{A}}^2 = \mathbf{a}^\top \mathbf{A} \mathbf{a}$. An augmented vector is denoted by $\underline{\mathbf{a}} = [\mathbf{a}_1, \dots, \mathbf{a}_i]^\top$. We denote linear transformations of sets by $\mathbf{A} \circ \mathcal{B} = \{\mathbf{A} \mathbf{b} \mid \mathbf{b} \in \mathcal{B}\}$ and $\mathcal{B} \circ \mathbf{A} = \{\mathbf{b} \mid \mathbf{A} \mathbf{b} \in \mathcal{B}\}$. The Cartesian product of the set \mathcal{A} and \mathcal{B} is $\mathcal{A} \times \mathcal{B} = \{[\mathbf{a}, \mathbf{b}] \mid \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}$. The n -ary Cartesian power of a set \mathcal{A} is denoted by $\mathcal{A}^n = \{[\mathbf{a}_1, \dots, \mathbf{a}_i] \mid \mathbf{a}_i \in \mathcal{A} \forall i \in \mathbb{I}_{1,n}\}$. The Minkowski sum of two sets is denoted $\mathcal{A} \oplus \mathcal{B} = \{\mathbf{a} + \mathbf{b} \mid \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}$ and the Pontryagin difference is given by $\mathcal{A} \ominus \mathcal{B} = \{\mathbf{c} \mid \mathbf{c} + \mathbf{b} \in \mathcal{A}, \forall \mathbf{b} \in \mathcal{B}\}$. A state at time step t is denoted by \mathbf{x}_t . Within an MPC optimal control problem, the state at prediction step k is denoted by \mathbf{x}_{k+t} , where the notation $\mathbf{x}_{k+t|t}$ explicitly denotes that the prediction \mathbf{x}_{k+t} was obtained at time step t . A function $\gamma: \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ is of class \mathcal{K} if it is strictly increasing and $\gamma(0) = 0$. A function $\alpha: \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ is of class \mathcal{K}_∞ if $\alpha \in \mathcal{K}$ and $\lim_{s \rightarrow \infty} \alpha(s) = \infty$. An asterisk denotes the optimal value, i.e., \mathbf{u}^* is the optimal solution.

II. PROBLEM SETUP

In the following, we first introduce the system dynamics and define properties of the uncertainty. Then, the MPC problem statement is introduced, including constraints for which violation probability must be minimized.

A. System Dynamics

We consider the linear, discrete-time dynamical control system

$$\mathbf{x}_{t+1} = \mathbf{A} \mathbf{x}_t + \mathbf{B} \mathbf{u}_t + \mathbf{G} \mathbf{w}_t \quad (1)$$

with state $\mathbf{x}_t \in \mathbb{R}^{n_x}$ and input $\mathbf{u}_t \in \mathbb{R}^{n_u}$ at time step t , as well as the bounded uncertainty $\mathbf{w}_t \in \mathcal{W} \subseteq \mathbb{R}^{n_w}$, where \mathbf{A}, \mathbf{B} have appropriate dimensions, $\mathbf{G} \in \mathbb{R}^{n_x \times n_w}$ is not singular.

Assumption 1: The uncertainty \mathbf{w}_t is a truncated Gaussian uncertainty with $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$, covariance matrix Σ_w , and bounded by the polytope \mathcal{W} .

In the following, MPC requires to predict the state trajectory. Based on the initial state \mathbf{x}_t , we denote the augmented system dynamics, delivering predictions $\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \dots, \mathbf{x}_{t+N}$, by

$$\underline{\mathbf{x}} = \underline{\mathbf{A}} \mathbf{x}_t + \underline{\mathbf{B}} \underline{\mathbf{u}} + \underline{\mathbf{G}} \underline{\mathbf{w}} \quad (2)$$

$$\text{where } \underline{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_{t+1} \\ \vdots \\ \mathbf{x}_{t+N} \end{bmatrix}, \quad \underline{\mathbf{u}} = \begin{bmatrix} \mathbf{u}_t \\ \vdots \\ \mathbf{u}_{t+N-1} \end{bmatrix}, \quad \underline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_t \\ \vdots \\ \mathbf{w}_{t+N-1} \end{bmatrix} \quad (3)$$

and $\underline{\mathbf{A}}, \underline{\mathbf{B}},$ and $\underline{\mathbf{G}}$ defined as in [35].

B. Model Predictive Control

MPC is applied to control system (1). In MPC, an optimal control problem is repeatedly solved on a finite horizon N , where state and input constraints are considered and only the first entry of the optimal input trajectory is applied. The prediction steps are denoted by index k . The MPC cost function is given by

$$J(\mathbf{x}_t, \underline{\mathbf{u}}) = \sum_{k=0}^{N-1} \left(\|\bar{\mathbf{x}}_{t+k}\|_{\mathbf{Q}}^2 + \|\bar{\mathbf{u}}_{t+k}\|_{\mathbf{R}}^2 \right) + \|\bar{\mathbf{x}}_{t+N}\|_{\mathbf{Q}_f}^2 \quad (4)$$

with weighting matrices \mathbf{Q}, \mathbf{R} and terminal weighting matrix \mathbf{Q}_f , where \mathbf{x}_t is known and $\bar{\mathbf{x}}_{t+k}$ denotes the mean of \mathbf{x}_{t+k} .

Furthermore, the set of admissible input sequences $\underline{\mathbf{u}}$ is given as the bounded polytopic set \mathcal{U}^N , which contains the origin. The polytopic set $\mathcal{X} \subset \mathbb{R}^{n_x}$ allows to formalize state constraints for all states in a prediction horizon of length N .

Assumption 2: The constraint set \mathcal{X} is closed, bounded, and contains the origin.

The constraint set \mathcal{X} may be expressed in augmented form by

$$\underline{x} \in \underline{\mathcal{X}} = \mathcal{X}^{N-1} \times \mathcal{X}_f. \quad (5)$$

A terminal constraint $\underline{x}_N \in \mathcal{X}_f \subseteq \mathcal{X}$ is introduced for the state \underline{x}_N . The set \mathcal{X}_f is defined later as part of the stability analysis.

C. Problem Statement

Both, RMPC and SMPC, do not provide adequate solutions if the probability of violating constraint (5) must be minimized. This problem may be formulated as

$$\min_{\underline{u} \in \mathcal{U}^N} \Pr(\underline{x} \notin \underline{\mathcal{X}}). \quad (6)$$

Given this constraint violation probability minimization, we now specify the problem to be addressed in this work.

Problem 1: The optimal control problem of each MPC iteration corresponding to time t is

$$\min_{\underline{u} \in \mathcal{U}^N} J(\underline{x}_t, \underline{u}) \quad (7a)$$

$$\text{s.t. } \underline{u} = \arg \min_{\underline{u} \in \mathcal{U}^N} \Pr(\underline{x} \notin \underline{\mathcal{X}}). \quad (7b)$$

In the following section, an MPC method is derived that provides a strategy to solve Problem 1.

III. METHOD

Here, we first present the CVPM-MPC method in Section III-A and then details on the probability optimization required for CVPM-MPC in Section III-B. Major properties of the CVPM-MPC method are discussed in Section IV.

A. CVPM-MPC

The general idea of CVPM-MPC is to solve an MPC optimal control problem where only those inputs are allowed that enable minimal constraint violation probability. Therefore, a set is defined that includes all inputs that minimize the constraint violation probability.

Definition 1 (Optimal CVPM Input Set): The optimal CVPM input set $\underline{\mathcal{U}}_{\text{cvpm}}$ consists of admissible input sequences \underline{u} that minimize the constraint violation probability $\Pr(\underline{x} \notin \underline{\mathcal{X}})$.

Determining $\underline{\mathcal{U}}_{\text{cvpm}}$ uses two cases, depending on whether an input sequence \underline{u} exists that guarantees constraint satisfaction or not.

Definition 2 (CVPM Safe Case): In the safe case, at least one admissible input sequence \underline{u} exists that guarantees constraint satisfaction, i.e.,

$$\exists \underline{u} \in \mathcal{U}^N \text{ s.t. } \Pr(\underline{x} \notin \underline{\mathcal{X}}) = 0. \quad (8)$$

Definition 3 (CVPM Probabilistic Case): In the probabilistic case, no input sequence \underline{u} exists that guarantees constraint satisfaction, i.e.,

$$\Pr(\underline{x} \notin \underline{\mathcal{X}}) > 0 \forall \underline{u} \in \mathcal{U}^N. \quad (9)$$

At each MPC iteration, it is determined whether the safe case is feasible, i.e. it is evaluated if the set of possible input sequences is not empty. If not, the probabilistic case is applied. In the following, for both cases separately, it is addressed how the set $\underline{\mathcal{U}}_{\text{cvpm}}$ is obtained.

1) Safe Case: Here the set $\underline{\mathcal{U}}_{\text{cvpm}}$ is

$$\underline{\mathcal{U}}_{\text{cvpm}} = \mathcal{U}^N \cap \left(\underline{\mathcal{X}} \ominus \left(\underline{\mathbf{G}} \circ \mathcal{W}^N \right) \oplus \{ -\underline{\mathbf{A}}\underline{x}_t \} \right) \circ \underline{\mathbf{B}}, \quad (10)$$

which is the intersection of the admissible input set \mathcal{U}^N and the set of input sequences that guarantee constraint satisfaction of (5). The set $\{ -\underline{\mathbf{A}}\underline{x}_t \}$ is a singleton that takes into account the current state \underline{x}_t and the set $\underline{\mathcal{U}}_{\text{cvpm}}$ is determined with algorithms from [35].

The intersection (10) does not only deliver the set of admissible inputs, it also allows to check if the safe case applies: if $\underline{\mathcal{U}}_{\text{cvpm}}$ is non-empty, an input sequence \underline{u} exists guaranteeing that the constraint (5) is satisfied.

Based on the previous result, it is possible to obtain a set of feasible initial states $\mathcal{X}_{\text{caseS}}$ (for the current MPC iteration) for which the safe case is applicable. This set is given by

$$\mathcal{X}_{\text{caseS}} = \left(\underline{\mathcal{X}} \ominus \left(\underline{\mathbf{G}} \circ \mathcal{W}^N \right) \oplus \left((-\underline{\mathbf{B}}) \circ \mathcal{U}^N \right) \right) \circ \underline{\mathbf{A}}, \quad (11)$$

which is obtained analog to (10). We make the following assumption to ensure that $\mathcal{X}_{\text{caseS}}$ is non-empty.

Assumption 3: The disturbances are small enough and propagated disturbances alone never exceed state constraints, i.e., $\underline{\mathbf{G}} \circ \mathcal{W}^N \subset \underline{\mathcal{X}}$. This assumption is straight-forward to satisfy when the system matrix \mathbf{A} is stable and motivates the following assumption.

Assumption 4: The system matrix \mathbf{A} is stable, i.e. the eigenvalues of \mathbf{A} are within the unit circle.

Remark 1: Note that even if the system is unstable, the assumption on \mathbf{A} can still be fulfilled by using a feedback controller for prestabilization. Then the sets \mathcal{X} and \mathcal{U} have to be redefined taking into account the prestabilization, such as in [36].

2) Probabilistic Case: For the safe case, $\underline{\mathcal{U}}_{\text{cvpm}}$ collects those input trajectories that guarantee constraint satisfaction. However, if such an input trajectory does not exist, at least minimal constraint violation probability can be ensured. Therefore, for the probabilistic case, $\underline{\mathcal{U}}_{\text{cvpm}}$ collects input trajectories \underline{u}^* that result in minimal constraint violation probability.

The safe case is applied if the current state is in $\mathcal{X}_{\text{caseS}}$ and results in zero constraint violation probability. Thus for all other states, i.e., $\underline{x}_t \notin \mathcal{X}_{\text{caseS}}$, the probability of constraint violation is non-zero. To minimize the probability of violating $\underline{x} \in \mathcal{X}^N$, we transform the problem to minimize the probability of violating $\underline{x} \in \mathcal{X}_{\text{caseS}}^N$

$$\underline{u}^* = \arg \min_{\underline{u} \in \mathcal{U}^N} \Pr(\underline{x} \notin \mathcal{X}_{\text{caseS}}^N). \quad (12)$$

The set $\mathcal{X} \setminus \mathcal{X}_{\text{caseS}}$ contains all states in \mathcal{X} that are not initial states for the safe case, where all predicted states remains in $\underline{\mathcal{X}}$. A trajectory will leave $\underline{\mathcal{X}}$ eventually if its initial state is in $\mathcal{X} \setminus \mathcal{X}_{\text{caseS}}$. Therefore, trajectories with states in $\mathcal{X} \setminus \mathcal{X}_{\text{caseS}}$ result in a high constraint violation probability. In contrast, $\mathcal{X}_{\text{caseS}} \setminus \mathcal{X}$ contains all states that are not in \mathcal{X} , but the predicted trajectory remains in $\underline{\mathcal{X}}$ and results in a zero probability of constraint violation in the subsequent MPC iterations. Changing the probability optimization from (7b) to (12) does, therefore, not significantly change \underline{u}^* . However, this adjustment makes it possible to prove stability (see Section IV-B). Once \underline{u}^* is obtained, we set:

$$\underline{\mathcal{U}}_{\text{cvpm}} = \{ \underline{u}^* \}. \quad (13)$$

3) CVPM-MPC Formulation: The CVPM-MPC optimal control problem is

$$\underline{u}^* = \arg \min_{\underline{u} \in \underline{\mathcal{U}}_{\text{cvpm}}} J(\underline{x}_t, \underline{u}) \quad (14)$$

with $\underline{\mathcal{U}}_{\text{cvpm}}$ according to (10) or (13). The closed-loop system, compare (1), is then given by

$$\underline{x}_{t+1} = \mathbf{A}\underline{x}_t + \mathbf{B}\underline{u}_t^* + \mathbf{G}\underline{w}_t \quad (15)$$

where \underline{u}_t^* is the first element of the optimal trajectory \underline{u}^* obtained at time step t .

Remark 2: Solving (14) requires two steps: first, a set $\underline{\mathcal{U}}_{\text{cvpm}}$ is calculated, and then, this set is used as a constraint for the MPC controller. Thus, CVPM can also serve as preprocessing for other controllers.

B. Probability Optimization in the Probabilistic Case

In the probabilistic case, the input trajectory \underline{u}^* with minimal constraint violation probability, is solution of the optimization problem (12). In general, neither analytic solution nor exact numerical solution of $\Pr(\underline{x} \notin \mathcal{X}_{\text{caseS}}^N)$ is possible. Therefore, we propose two approximations. In the first approximation, the probabilities are computed using a sampling-based approach, which allows to increase the accuracy as the number of samples increases. However, this computation is time-consuming and not suitable for fast real-time systems. Therefore, the second method does not approximate the probability but modifies the problem to find the input sequence \underline{u} .

In the following, we prepare both approximations. We assume, the probability distribution in the safe case has a truncated support. This ensures existence of trajectories with a zero probability of constraint violation. However, in the probabilistic case, this leads to a vanishing gradient in the optimization. Therefore, we approximate the truncated Gaussian of Assumption 1 by the corresponding non-truncated Gaussian $\underline{w}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$. The mean $\bar{\underline{x}}$ of the state trajectory \underline{x} is $\bar{\underline{x}} = \underline{A}\underline{x}_t + \underline{B}\underline{u}$ and the covariance matrix is given as $\Sigma_{\underline{x}} = \text{diag}(\Sigma_x, \dots, \Sigma_x)$, where Σ_x is the steady-state solution of the uncertainty propagation

$$\Sigma_x = \underline{A}\Sigma_x\underline{A}^\top + \underline{G}\Sigma_w\underline{G}^\top. \quad (16)$$

Given the mean and the covariance matrix for the state sequence \underline{x} with a particular input sequence \underline{u} , we obtain

$$\underline{x} \sim \mathcal{N}(\bar{\underline{x}}, \Sigma_{\underline{x}}) = \mathcal{N}(\underline{A}\underline{x}_t + \underline{B}\underline{u}, \Sigma_{\underline{x}}), \quad (17)$$

proving that the state trajectory is subject to a Gaussian distribution.

1) *Sampling-Based Probability Optimization*: A numerical Monte Carlo sampling approach is employed to determine the probability of violating constraints. From the distribution (17), N_s samples of state sequences are drawn. The number of samples that are an element of the set $\underline{\mathcal{X}}$ is denoted as $N_{\underline{\mathcal{X}}}$. It follows that the constraint violation probability is approximately

$$\Pr(\underline{x} \notin \underline{\mathcal{X}}) \approx 1 - \frac{N_{\underline{\mathcal{X}}}}{N_s}. \quad (18)$$

The minimization of the constraint violation probability is approximated with a numeric optimization of (18). In each step within the optimization, (18) must be determined for a given input sequence \underline{u} resulting in a huge computational burden.

2) *Probability Optimization approximated as Quadratic Program*: We propose an approximation in the following to speed up the computation. For the probabilistic case, the input sequence \underline{u}^* is defined as the solution to the optimization problem (12) with

$$\Pr(\underline{x} \notin \mathcal{X}_{\text{caseS}}^N) = 1 - c \int_{\mathcal{X}_{\text{caseS}}^N} e^{-\frac{1}{2}(\bar{\underline{x}} - \underline{\xi})^\top \Sigma_{\underline{x}}^{-1}(\bar{\underline{x}} - \underline{\xi})} d\underline{\xi}, \quad (19)$$

where $c = \left((2\pi)^{n_x N} \det \Sigma_{\underline{x}} \right)^{-1}$. As (19) is a non-convex cost for the optimization problem, we approximate it. In the probabilistic case, the mean state sequence $\bar{\underline{x}}$ is not in $\mathcal{X}_{\text{caseS}}^N$, otherwise the safe case is applied. Therefore, we assume that the distribution function (17) is nearly constant over the set $\mathcal{X}_{\text{caseS}}$. We then approximate the integral in (19) by a multiplication of the probability density function for the states in (17) and the volume of the polytope $V_P(\mathcal{X}_{\text{caseS}}^N)$, yielding

$$\Pr(\underline{x} \notin \mathcal{X}_{\text{caseS}}^N) \approx 1 - c e^{-\frac{1}{2}(\bar{\underline{x}} - \underline{\xi})^\top \Sigma_{\underline{x}}^{-1}(\bar{\underline{x}} - \underline{\xi})} V_P(\mathcal{X}_{\text{caseS}}^N). \quad (20)$$

The probability density function is evaluated at a point $\underline{\xi} \in \mathcal{X}_{\text{caseS}}^N$ within the polytope. The variable $\underline{\xi}$ is then included in the optimization, yielding a similar structure as the MPC cost function. Later,

this structure is essential for the proof of stability. The volume of the polytope $V_P(\mathcal{X}_{\text{caseS}})$ does not change if \underline{u} is varied. We need to minimize $\Pr(\underline{x} \notin \mathcal{X}_{\text{caseS}}^N)$, thus it is sufficient to solve the quadratic optimization problem

$$\underline{u}^* = \arg \min_{\underline{u} \in \mathcal{U}^N, \underline{\xi} \in \mathcal{X}_{\text{caseS}}^N} (\bar{\underline{x}} - \underline{\xi})^\top \tilde{\Sigma}_{\underline{x}}^{-1} (\bar{\underline{x}} - \underline{\xi}) \quad (21a)$$

$$\text{s.t. } \bar{\underline{x}} = \underline{A}\underline{x}_t + \underline{B}\underline{u}. \quad (21b)$$

The matrix $\tilde{\Sigma}_{\underline{x}}$ is an adapted version of $\Sigma_{\underline{x}}$ and is defined later when stability is discussed. The solution of (21) leads to state sequences in the direction of small eigenvalues of $\tilde{\Sigma}_{\underline{x}}^{-1}$, which result in the fastest decrease of the probability of constraint violation.

IV. PROPERTIES

A. Recursive Feasibility

Recursive feasibility is a fundamental property for MPC algorithms. In each MPC time step, an optimal control problem has to be solved, and it needs to be ensured that the optimal control problem is feasible at time step $t+1$ if it is feasible at time step t .

Definition 4 (Recursive Feasibility): An MPC optimal control problem is recursively feasible if it holds that $\underline{\mathcal{U}}_t \neq \emptyset \rightarrow \underline{\mathcal{U}}_{t+1} \neq \emptyset$ for all $t \in \mathbb{N}_0$ where $\underline{\mathcal{U}}_t$ is the set of admissible inputs at time step t .

Recursive feasibility is fulfilled in CVPM per construction, because if the safe case is infeasible, the probabilistic case is applied that will always deliver a solution. In the following, we prove recursive feasibility of the safe case, i.e., once the safe case is applicable, $\mathcal{X}_{\text{caseS}}$ is invariant under CVPM-MPC. For this purpose, the following assumption is required, which defines the terminal set in (5).

Assumption 5: The terminal set $\mathcal{X}_f \subseteq \mathcal{X}$ is chosen as a robust control invariant set [35] and fulfills

$$\underline{A} \circ \mathcal{X}_f \oplus \underline{A}^N \underline{G} \circ \mathcal{W} \subseteq \mathcal{X}_f \oplus (-\underline{B}) \circ \mathcal{U}. \quad (22)$$

Remark 3: Ass. 5 yields candidates for the terminal set \mathcal{X}_f by extending the standard notion for a robust invariant set $\underline{A} \circ \mathcal{X}_f \subseteq \mathcal{X}_f \oplus \underline{G} \circ \mathcal{W} \oplus (-\underline{B}) \circ \mathcal{U}$ [35], such that disturbances propagated over the horizon N are included, i.e. it is a robust control invariant set where the set of disturbances is $\underline{A}^N \underline{G} \circ \mathcal{W}$.

Lemma 1: If Assumption 5 holds, then for all $\underline{x}_t \in \mathcal{X}_{\text{caseS}}$ there exists an input \underline{u}_t such that the state at the next time step is also in the set $\mathcal{X}_{\text{caseS}}$, i.e., $\mathcal{X}_{\text{caseS}}$ is robust control invariant.

Proof: The proof is given in Appendix I. ■

Based on Lemma 1, we can now formulate the following theorem on recursive feasibility.

Theorem 1: The safe case is recursively feasible.

Remark 4: In a practical situation, a terminal set, not satisfying Assumption 5 is also possible. This would imply that $\mathcal{X}_{\text{caseS}}$ is not robust control invariant and thus the recursive feasibility guarantee in the safe case is lost. Nevertheless, still a solution with possibly non zero probability of constraint violation can be found.

B. Stability

We show that both the safe case and the probabilistic case ensure input-to-state stability, proving that the CVPM-MPC method is input-to-state stable (ISS). We start with the definition of an ISS Lyapunov function.

Definition 5 (ISS Lyapunov Function [37]): Consider a continuous function $V : \mathcal{X}_{\text{rpi}} \rightarrow \mathbb{R}$ and the functions $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$, $\gamma \in \mathcal{K}$. Then, V is an ISS Lyapunov function for a system $\underline{x}_{t+1} = \underline{f}(\underline{x}_t, \underline{w}_t)$, where \underline{f} is continuous, with a robust positively invariant set \mathcal{X}_{rpi} if $\alpha_1, \alpha_2, \alpha_3, \gamma$ exist, for all $k \geq 0$ and $\underline{x}_t \in \mathcal{X}_{\text{rpi}}$, such that

$$\alpha_1(\|\mathbf{x}_t\|) \leq V(\mathbf{x}_t) \leq \alpha_2(\|\mathbf{x}_t\|) \quad (23a)$$

$$V(\mathbf{f}(\mathbf{x}_t, \mathbf{w}_t)) - V(\mathbf{x}_t) \leq -\alpha_3(\|\mathbf{x}_t\|) + \gamma(\|\mathbf{w}_t\|). \quad (23b)$$

Lemma 2 (Sufficient ISS Condition [37]): The origin of system $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{w}_t)$ where f is continuous is ISS if an ISS Lyapunov function according to Definition 5 exists.

The following assumption is required.

Assumption 6: The weighting matrices of the stage cost are positive definite and symmetric, i.e., $\mathbf{Q} = \mathbf{Q}^\top \succ 0$ and $\mathbf{R} = \mathbf{R}^\top \succ 0$. The terminal cost weighting matrix \mathbf{Q}_f is a solution of the discrete-time algebraic Riccati equation.

Based on the robust control invariance of $\mathcal{X}_{\text{caseS}}$ it can be shown that the origin is ISS for the safe case.

Lemma 3: Let Assumptions 2, 4, 5 and 6 hold. Then, for $\mathbf{x}_t \in \mathcal{X}_{\text{caseS}}$, the origin of the closed-loop system (15) is ISS.

Proof: The proof is given in Appendix II. ■

Consecutively applying the safe case yields same behavior as robust MPC. If the safe case is not applicable, i.e., $\mathbf{x}_t \notin \mathcal{X}_{\text{caseS}}$, it needs to be ensured that the system is still ISS. For the inverse covariance matrix of the last predicted state \mathbf{x}_N , the solution \mathbf{S} of the discrete-time algebraic Riccati equation is used, i.e.,

$$(\mathbf{A} - \mathbf{BK})^\top \mathbf{S}(\mathbf{A} - \mathbf{BK}) + \Sigma_{\mathbf{x}}^{-1} = \mathbf{S} \quad (24)$$

where \mathbf{K} is a control gain such that $\mathbf{x}_t \in \mathcal{X}_{\text{caseS}} \Rightarrow \mathbf{x}_{t+1} \in \mathcal{X}_{\text{caseS}}$. It follows that

$$\tilde{\Sigma}_{\mathbf{x}}^{-1} = \text{diag}(\Sigma_{\mathbf{x}}^{-1}, \dots, \Sigma_{\mathbf{x}}^{-1}, \mathbf{S}) \quad (25)$$

describes the adapted inverse of the covariance matrix.

Lemma 4: Let Assumptions 4 and 6 hold. For $\mathbf{x}_t \notin \mathcal{X}_{\text{caseS}}$, the origin of the error dynamics of the error between the closed-loop system (15) and the optimization variable $\underline{\xi} \in \mathcal{X}_{\text{caseS}}$ is ISS using the method from Section III-B.2.

Proof: The proof is given in Appendix III. ■

Based on Lemma 3 and Lemma 4, we can now formulate the stability theorem for CVPM-MPC.

Theorem 2: Let Assumptions 2, 4, 5 and 6 hold. The origin of the closed-loop system (15), controlled by CVPM-MPC, is ISS.

Proof: From Lemma 4 we conclude that the state converges to the set $\mathcal{X}_{\text{caseS}}$. Based on Lemma 3 the origin of system (15) is ISS for $\mathbf{x}_t \in \mathcal{X}_{\text{caseS}}$. Finally, the origin is ISS for all $\mathbf{x}_t \in \mathbb{R}^{n_x}$. ■

C. Extension to Time-Variant Constraints

We have assumed that the state constraint set \mathcal{X} and the disturbance set \mathcal{W} are time-invariant. However, the method can also handle time-variant \mathcal{X} and \mathcal{W} during runtime without preprocessing. We discuss two different situations here. First, we discuss stability when the sets \mathcal{X} and \mathcal{W} change for a short time, and second, when the sets change permanently.

1) *Short-term change:* A change of the sets \mathcal{X} or \mathcal{W} for one time step, i.e., an unexpected large disturbance, may lead to a violation of Assumptions 2 and 5, and thus invalidate the stability guarantee in the safe case. A violation of Assumption 3 leads to $\mathcal{X}_{\text{caseS}} = \emptyset$ and, therefore, the probabilistic case is applied. Although recursive feasibility is preserved, stability of the origin can no longer be ensured. As the temporary change subsides, all assumptions are again satisfied, resulting in stability.

2) *Permanent change:* It can be shown that, under Assumptions 2, 3, and 5, also a permanent change of the sets \mathcal{X} and \mathcal{W} does not affect the stability of the system.

To meet Assumption 5, a precomputed terminal set \mathcal{X}_f can be utilized, which is computed for an initially large disturbance set \mathcal{W} . If the disturbance set changes during runtime, the assumptions still hold. A suitable choice for \mathcal{X}_f is a minimal robust invariant set. This,

due to $\mathcal{X}_f \subseteq \mathcal{X}$, enables to choose the state constraint set \mathcal{X} rather small.

V. NUMERICAL EXAMPLE

In the following, we discuss recursive feasibility and stability of the proposed method in simulation studies and demonstrate the capability of CVPM-MPC to handle time-variant uncertainty bounds.

A. Simulation Setup

We consider a discrete-time linear system. The system matrix, input matrix, and disturbance matrix are given by

$$\mathbf{A} = \begin{bmatrix} 0.99 & -0.02 \\ 0.21 & 0.92 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0.30 \\ 0.06 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 0.02 & 0.00 \\ 0.01 & 0.19 \end{bmatrix}. \quad (26)$$

The model describes a DC-DC converter, see [38], where the state x_1 stands for the current in a coil and the state x_2 is the voltage of a capacitor. The goal is to stabilize a voltage of 3.3 V, yielding the reference $\mathbf{x}_{\text{ref}} = [1.06 \quad 3.30]^\top$, and $\mathbf{u}_{\text{ref}} = 0.28$. The method is adapted such that the reference state is stabilized. The input is the duty cycle of a transistor, thus $\mathcal{U} = \{\mathbf{u} \mid 0 \leq \mathbf{u} \leq 1\}$. In the simulation, modeled and unmodeled disturbances are considered. The support of the modeled disturbance is

$$\mathcal{W} = \left\{ [w_1, w_2]^\top \mid -0.2 \leq w_1 \leq 0.2, -0.2 \leq w_2 \leq 0.2 \right\} \quad (27)$$

with covariance matrix $\Sigma_{\mathbf{w}} = \text{diag}(0.2, 0.2)$. The time-invariant state constraint set is chosen as

$$\mathcal{X} = \left\{ [x_1, x_2]^\top \mid 0 \leq x_1 \leq 2, 2.8 \leq x_2 \leq 3.8 \right\}. \quad (28)$$

Note that state constraints, uncertainty, and system matrix satisfy Assumptions 1, 2 and 4.

The MPC employs a horizon of $N = 10$ with sampling time $\Delta t = 0.1$ and the weighting matrices are chosen as $\mathbf{Q} = \text{diag}(1, 5)$, and $\mathbf{R} = 1$. The terminal cost \mathbf{Q}_f is determined according to Assumption 6. The computation of the polyhedra $\underline{\mathcal{U}}_{\text{cvpm}}$ and $\mathcal{X}_{\text{caseS}}$ is done with the MPT3 toolbox [39].

B. Comparison of Probability Minimization Methods

The minimum of the constraint violation probability (12) is challenging to compute. Therefore, two methods to approximate the probability are introduced. First, in Section III-B.1, a numeric computation of the probability using a Monte Carlo method, and in Section III-B.2, an approximation of the probability utilizing a quadratic program.

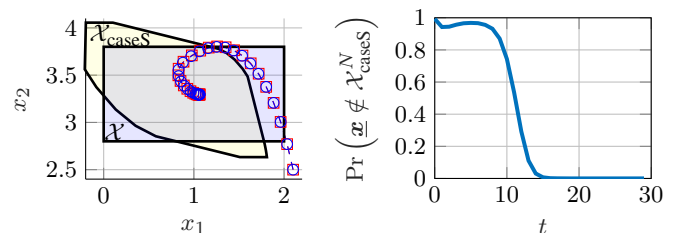


Fig. 1: Comparison of probability minimization methods; Left: trajectory of Monte Carlo Integration (red), Quadratic Program approximation (blue, Sec. III-B.2); Right: constraint violation probability

As shown on the left in Figure 1, the trajectories that result from applying both methods are almost identical. The initial value is not in \mathcal{X} leading to a high probability of constraint violation. Applying the proposed method leads to a convergence to the set \mathcal{X} and a decrease

in the constraint violation probability. On the right side in Figure 1, the constraint violation probability for the approach from Section III-B.2 is given. Initially the constraint violation probability is almost 1 and the minimization allows to approach the set $\mathcal{X}_{\text{caseS}}$. We observe that the probabilistic case can deal with otherwise infeasible initial states and find a trajectory that converges to $\mathcal{X}_{\text{caseS}}$.

The average computation time for one step of the approximation introduced in Section III-B.2 is 60 ms on a standard computer while the average computation time of one step of the sampling method is 4 min on a computer with an Intel Xeon E5-2630. The sampling-based approach uses $N_s = 10^5$ randomly generated samples in each iteration distributed according to (17).

C. Performance with Modeled and Unmodeled Disturbances

The following simulation shows convergence to a reference when starting with a non-zero constraint violation probability. At time step $t = 50$, an unmodeled disturbance affects the system for one time step, which is handled by the CVPM-MPC method. An unmodeled disturbance may be interpreted as an increase of \mathcal{W} for one time step or as $w_t \notin \mathcal{W}$. Figure 2 illustrates the simulation results.

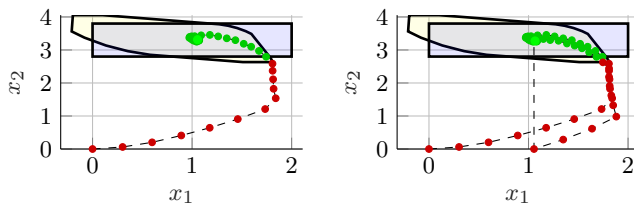


Fig. 2: Simulation of CVPM-MPC with an unmodeled disturbance: Left: convergence to the set \mathcal{X} ; Right: convergence to the set \mathcal{X} after unmodeled disturbance. (green: safe case; red: probabilistic case; bright marker: current state)

The set \mathcal{X} is indicated by the blue box. The set $\mathcal{X}_{\text{caseS}}$ is marked in yellow. If the system state is in $\mathcal{X}_{\text{caseS}}$, the safe case is applicable. The initial state does not allow zero constraint violation probability in the next step, i.e., $x_0 \notin \mathcal{X}_{\text{caseS}}$; therefore, the probabilistic case is required, indicated by the red dot in Figure 2.

Applying the CVPM-MPC procedure for the probabilistic case moves the system state into the set $\mathcal{X}_{\text{caseS}}$, as seen on the left side in Figure 2. In $\mathcal{X}_{\text{caseS}}$ the control input is determined based on the safe case, as indicated by the green dots since it is possible to reach \mathcal{X} in the next step. The subsequent steps with the safe case move the system state towards the origin. Not that in this simulation, mostly the probabilistic case is active whereas in application the standard situation is applying the safe case and only switching to the probabilistic case when unexpected disturbances occur.

At time step $t = 50$, an unmodeled disturbance occurs, which moves the system state outside of $\mathcal{X}_{\text{caseS}}$, as illustrated on the right side in Figure 2. Note that input-to-state stability is not guaranteed in this step as the uncertainty bound increased which violates Assumption 1. Similar to the initial simulation state, the probabilistic case is required because it is not possible to reach the constraint set \mathcal{X} in the next step. By switching from the safe case to the probabilistic case, recursive feasibility is maintained. Afterwards, the CVPM-MPC method steers the system state back to $\mathcal{X}_{\text{caseS}}$.

VI. DISCUSSION

In contrast to the CVPM-MPC method presented in [5] and [6], the CVPM-MPC approach proposed in this work is more general. Here, we consider general linear constraints for a system with additive

uncertainty, whereas [5], [6] are motivated by a vehicle collision avoidance scenario utilizing two dynamics and norm constraints. The proposed approach significantly extends the possible applications to all linear or linearized systems where constraints are linear or where constraints may be linearized. The stability of the origin with general linear constraints is achieved with a robust control invariant set of initial states for the safe case and Assumption 5 on the terminal constraint.

In Section IV-B, stability is discussed. As seen in the simulation example, the CVPM-MPC method is capable of remaining feasible even in the presence of unmodeled disturbances, but at the cost of a constraint violation probability close to 1. The stability results may not hold in case of unmodeled disturbances, as a bounded uncertainty is assumed in the proofs (Assumption 1). The proposed method, however, allows updating the assumed uncertainty bound and state constraint. Therefore, the stability proof becomes valid again for an updated set \mathcal{W} or \mathcal{X} as long as Assumptions 2, 3, and 5 hold. If the uncertainty bound is not known initially, a conservative guess may be chosen and then the bound may be tightened over time, based on recorded data. The potential short loss of a stability guarantee is acceptable, however, as the main focus of this work is the minimization of constraint violation probability. Note that recursive feasibility remains guaranteed even for unmodeled disturbances.

If only SMPC is applied to the scenario shown in Fig. 2, an 8 % chance of violating the $x_1 < 2$ constraint is observed, while CVPM avoids constraint violation robustly. In contrast to SMPC, the probabilistic case does not optimize a control objective but focuses only on the constraint by minimizing the probability of constraint violation. Only when the measured state, which is initial value for the prediction, is in $\mathcal{X}_{\text{caseS}}$, the control objective is again taken into account. We assume to have applications where the safe case is active almost all the time while the probabilistic case tackles rare problems with safety and is active if initial values are not safe or unmodeled disturbances occur. In contrast to RMPC, we consider the disturbance set \mathcal{W} as a tuning parameter. Conservative assumptions with large \mathcal{W} results in conservative policies, whereas a small \mathcal{W} leads to a more optimistic but also more risky behavior.

VII. CONCLUSION

The proposed CVPM-MPC method provides an MPC approach that combines the advantages of robust and stochastic MPC. The ability of CVPM-MPC to cope with time-variant constraints and uncertainty bounds provides a significant benefit for safety-critical systems. Recursive feasibility is guaranteed and stability is ensured under assumptions on the state constraint and disturbance support.

CVPM-MPC is suitable for linear and linearized systems, enabling the use in applications such as quadcopter control or automated vehicles. Furthermore, the proposed CVPM-MPC method may be extended to consider probabilistic constraints and robust state constraints simultaneously. This extension would allow practitioners to employ robust constraints where possible and necessary as well as probabilistic CVPM constraints if suitable.

Whereas robust methods guarantee safety for predictable events, unpredictable environmental changes are not covered. This is especially complex if ethical concerns are relevant for applications, e.g., how autonomous systems should behave if collision avoidance cannot be guaranteed. CVPM-MPC provides a novel way to handle such scenarios and ethical issues.

APPENDIX I PROOF OF LEMMA 1

In this section, we use the following notation. The input sequence \underline{u}_t obtained at time step t until prediction step $t + N - 1$ yields the

state sequence \underline{x}_t , where these sequences are defined as

$$\underline{u}_t = (\mathbf{u}_{t|t}, \dots, \mathbf{u}_{t+N-1|t}) \text{ and } \underline{x}_t = (\mathbf{x}_{t+1|t}, \dots, \mathbf{x}_{t+N|t}). \quad (29)$$

The initial state for the state sequence is given as $\mathbf{x}_{t|t} = \mathbf{x}_t$. Predictions made at time step $t+1$ are denoted

$$\mathbf{x}_{t+1|t+1} = \mathbf{A}\mathbf{x}_{t|t} + \mathbf{B}\mathbf{u}_{t|t} + \mathbf{G}\mathbf{w}_t = \mathbf{x}_{t+1|t} + \mathbf{G}\mathbf{w}_t, \quad (30)$$

depending on the uncertainty \mathbf{w}_t at time step t . Optimal trajectories for the states and inputs are denoted by \underline{u}_t^* and \underline{x}_t^* .

Proof: We first show that a subsequent state sequence is in the zero violation set $\underline{\mathcal{X}} \ominus (\underline{\mathbf{G}} \circ \mathcal{W}^N)$ if the previous state sequence is also in the zero violation set.

Let the predicted state sequence $(\mathbf{x}_{t+1|t} \dots \mathbf{x}_{t+N|t})$, based on the initial state $\mathbf{x}_t \in \mathcal{X}_{\text{caseS}}$, be in $\underline{\mathcal{X}} \ominus (\underline{\mathbf{G}} \circ \mathcal{W}^N)$. It follows that

$$\mathbf{x}_{t+i|t} \in \mathcal{X} \ominus \mathbf{A}^{i-1}\mathbf{G} \circ \mathcal{W} \ominus \mathbf{A}^{i-2}\mathbf{G} \circ \mathcal{W} \ominus \dots \ominus \mathbf{G} \circ \mathcal{W}. \quad (31)$$

Then, the subsequent state sequence initial state $\mathbf{x}_{t+1|t+1} = \mathbf{x}_{t+1|t} + \mathbf{G}\mathbf{w}_t$ is affected by the disturbance \mathbf{w}_t . All states in the candidate state sequence are affected by the propagation of the disturbance yielding

$$\begin{bmatrix} \mathbf{x}_{t+2|t+1} \\ \vdots \\ \mathbf{x}_{t+N|t+1} \\ \mathbf{x}_{t+N+1|t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{t+2|t} \\ \vdots \\ \mathbf{x}_{t+N|t} \end{bmatrix} + \begin{bmatrix} \mathbf{A}\mathbf{G} \\ \vdots \\ \mathbf{A}^{N-1}\mathbf{G} \\ \mathbf{A}^N\mathbf{G} \end{bmatrix} \mathbf{w}_t. \quad (32)$$

For a disturbance $\mathbf{w}_t \in \mathcal{W}$ the $t+i$ -th state $\mathbf{x}_{t+i|t+1}$ is in

$$\mathcal{X} \ominus \mathbf{A}^{i-1}\mathbf{G} \circ \mathcal{W} \ominus \dots \ominus \mathbf{G} \circ \mathcal{W} \oplus \mathbf{A}^{i-1}\mathbf{G} \circ \mathcal{W} \quad (33a)$$

$$= \mathcal{X} \ominus \mathbf{A}^{i-2}\mathbf{G} \circ \mathcal{W} \ominus \dots \ominus \mathbf{G} \circ \mathcal{W} \quad \forall i \in \mathbb{I}_{2,N}. \quad (33b)$$

Next, we show that the terminal state does not leave the terminal set. The state $\mathbf{x}_{t+N|t}$ is in the terminal set \mathcal{X}_f . Based on Assumption 5, it follows that an input \mathbf{u} exists such that the terminal state of the subsequent sequence is also in the terminal set, i.e., $\mathbf{x}_{t+N+1|t+1} \in \mathcal{X}_f$. To conclude, the subsequent state sequence (32) is always in the zero violation set $\underline{\mathcal{X}} \ominus (\underline{\mathbf{G}} \circ \mathcal{W}^N)$.

Since $\mathcal{X}_{\text{caseS}}$ includes all states where a prediction $\underline{x}_t \in \underline{\mathcal{X}} \ominus (\underline{\mathbf{G}} \circ \mathcal{W}^N)$ exists and the subsequent prediction \underline{x}_{t+1} affected by a disturbance $\mathbf{w}_t \in \mathcal{W}$ is in the same set, we conclude that $\mathcal{X}_{\text{caseS}}$ is a robust control invariant set. ■

APPENDIX II PROOF OF LEMMA 3

We start with preparations for the proof. The optimal input $\mathbf{u}_t^*(\mathbf{x}_t)$ is obtained by solving the MPC optimal control problem (14). Based on (4), we abbreviate the stage cost and the terminal cost by $l(\mathbf{x}_t, \mathbf{u}_t) = \|\mathbf{x}_t\|_{\mathbf{Q}}^2 + \|\mathbf{u}_t\|_{\mathbf{R}}^2$, and $V_f(\mathbf{x}_t) = \|\mathbf{x}_t\|_{\mathbf{Q}_f}^2$. In the following, the notation introduced in (29) is used.

Proof: For $\mathbf{x}_{t|t} \in \mathcal{X}_{\text{caseS}}$ with Assumption 5, $\mathcal{X}_{\text{caseS}}$ is robust control invariant according to Lemma 1. We define the Lyapunov function $V(\mathbf{x}_{t|t}) = J(\mathbf{x}_{t|t}, \underline{\mathbf{u}}_t^*)$ based on the cost (4) with the optimal feedback law $\underline{\mathbf{u}}_t^*(\mathbf{x}_t)$ obtained according to (14) where $\underline{\mathcal{U}}_{\text{cvpm}}$ is from the the safe case. As V is continuous, positive definite, and radially unbounded based on Assumption 6, $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ exist such that (23a) is fulfilled [40]. Additionally, V is Lipschitz continuous on $\mathcal{X}_{\text{caseS}}$ as V only consists of quadratic terms and $\mathcal{X}_{\text{caseS}}$ is bounded.

Next, we prove the descent property (23b). Based on $\underline{\mathbf{u}}_t^* = (\mathbf{u}_{t|t}^*, \dots, \mathbf{u}_{t+N-1|t}^*)$ we obtain

$$V(\mathbf{x}_{t|t}) = l(\mathbf{x}_{t|t}, \mathbf{u}_{t|t}^*) + q(\mathbf{x}_{t+1|t}^*) \quad (34)$$

where $q(\mathbf{x}_{t+1|t}^*) = \sum_{k=1}^{N-1} l(\mathbf{x}_{t+k|t}^*, \mathbf{u}_{t+k|t}^*) + V_f(\mathbf{x}_{t+N|t}^*)$ summarizes the total cost starting at $\mathbf{x}_{t+1|t}^*$, which is used similarly in [41]. For $t+1$ with $\underline{\mathbf{u}}_{t+1}^*$ the optimal cost is $J(\mathbf{x}_{t+1|t+1}, \underline{\mathbf{u}}_{t+1}^*)$ and with the non-optimal input sequence $(\mathbf{u}_{t+1|t}^*, \dots, \mathbf{u}_{t+N-1|t}^*, \tilde{\mathbf{u}})$, we obtain due to optimality an input $\tilde{\mathbf{u}}$ satisfying Assumption 6

$$\begin{aligned} V(\mathbf{x}_{t+1|t+1}) &\leq \sum_{k=1}^N l(\mathbf{x}_{t+k|t+1}, \mathbf{u}_{t+k|t}^*) + V_f(\mathbf{x}_{t+N+1|t+1}) \quad (35a) \\ &= q(\mathbf{x}_{t+1|t+1}) \quad (35b) \end{aligned}$$

where $q(\mathbf{x}_{t+1|t+1}) = q(\mathbf{x}_{t+1|t}^* + \mathbf{G}\mathbf{w}_t)$ according to (30). Note that at $t+1$ the optimal input at prediction step k is $\mathbf{u}_{t+k|t+1}^*$, based on $\mathbf{x}_{t+k|t+1}$, whereas $\mathbf{u}_{t+k|t}^*$, obtained at t , is not optimal as $\mathbf{x}_{t+1|t+1}$ was affected by \mathbf{w}_t . With (34) and (35) we obtain

$$V(\mathbf{x}_{t+1|t+1}) - V(\mathbf{x}_{t|t}) \quad (36a)$$

$$\leq q(\mathbf{x}_{t+1|t+1}) - l(\mathbf{x}_{t|t}, \mathbf{u}_{t|t}^*) - q(\mathbf{x}_{t+1|t}^*). \quad (36b)$$

The term $q(\cdot)$ is Lipschitz continuous due to Assumption 6 and the boundedness of $\mathcal{X}_{\text{caseS}}$ and \mathcal{U} , resulting in

$$\left\| q(\mathbf{x}_{t+1|t}^* + \mathbf{G}\mathbf{w}_t) - q(\mathbf{x}_{t+1|t}^*) \right\| \leq L_q \|\mathbf{G}\mathbf{w}_t\| \quad (37)$$

with Lipschitz constant L_q . Given (36b) and (37),

$$V(\mathbf{x}_{t+1|t+1}) - V(\mathbf{x}_{t|t}) \leq -l(\mathbf{x}_{t|t}, \mathbf{u}_{t|t}^*) + L_q \|\mathbf{G}\mathbf{w}_t\| \quad (38a)$$

$$\leq -\alpha_3 (\|\mathbf{x}_{t|t}\|) + L_q \|\mathbf{G}\mathbf{w}_t\|. \quad (38b)$$

It is straightforward that the previous procedure holds for any $t \geq 0$. Therefore, all requirements of Definition 5 are fulfilled, i.e., V is an ISS Lyapunov function and the origin of system (15) is ISS if $\mathbf{x}_{t|t} \in \mathcal{X}_{\text{caseS}}$. ■

APPENDIX III PROOF OF LEMMA 4

In the following, the notation introduced in (29) is used. *Proof:* The optimal input is now determined in (21). We define the error between a state sequence \underline{x}_k and the optimization variable $\underline{\xi}$ in each time step $t+k$ as $\mathbf{e}_{t+k|t}$ based on the initial value at time t . The candidate Lyapunov function is

$$V'(\mathbf{e}_{t|t}) = \sum_{k=0}^{N-1} l'(\mathbf{e}_{t+k|t}^*) + V_f'(\mathbf{e}_{t+N|t}^*) \quad (39a)$$

$$= l'(\mathbf{e}_{t|t}^*) + q'(\mathbf{e}_{t+1|t}^*) \quad (39b)$$

with an optimal input sequence $\underline{\mathbf{u}}_k^*$, values of $\underline{\xi}_k^*$ and stage cost $l'(\mathbf{e}_{t+k|t}^*) = \|\mathbf{e}_{t+k|t}^*\|_{\Sigma_{-1}}^2$ according to (21). The terminal cost is defined as $V_f'(\mathbf{e}_{t+N|t}^*) = \|\mathbf{e}_{t+N|t}^*\|_{\mathbf{S}}^2$, based on (25). Since V' is positive definite, radially unbounded, and Lipschitz continuous on \mathbb{R}^{n_x} , (23a) holds.

Similar to Lemma 3, we define the total cost starting at $\mathbf{e}_{t+1|t}$ as $q'(\mathbf{e}_{t+1|t}^*)$. For $t+1$ the optimal cost is $V'(\mathbf{e}_{t+1|t+1})$ with the optimal values $\underline{\mathbf{u}}_{k+1}^*$ and $\underline{\xi}_{k+1}^*$. We apply a shifted non-optimal input sequence $(\mathbf{u}_{t+1|t}^*, \dots, \mathbf{u}_{t+N-1|t}^*, \mathbf{K}\mathbf{x}_{t+N-1|t}^*)$, where the last input is determined with the feedback matrix \mathbf{K} . Similar for the error, the shifted sequence $(\mathbf{e}_{t+1|t}, \dots, \mathbf{e}_{t+N|t}, (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{e}_{t+N|t})$ is used. We obtain an upper bound

$$V'(\mathbf{e}_{t+1|t+1}) \leq \sum_{k=1}^N l'(\mathbf{e}_{t+k|t+1}^*) + V_f'(\mathbf{e}_{t+N+1|t+1}^*) \quad (40a)$$

$$= q'(\mathbf{e}_{t+1|t+1}^*), \quad (40b)$$

which can be summarized to $q'(e_{t+1|t+1})$ due to (25). Since q' is Lipschitz continuous on \mathbb{R}^{n_x} , the same arguments as in Lemma 3 yields

$$V'(e_{t+1|t+1}) - V'(e_{t|t}) \leq -\alpha_3 \left(\|e_{t|t}^*\| \right) + L_{q'} \|Gw_t\|. \quad (41)$$

It follows that for $x_{t|t} \in \mathbb{R}^{n_x}$, which includes $x_{t|t} \notin \mathcal{X}_{\text{caseS}}$, the error $e_{t|t}$ between (15) and the optimization variable $\underline{x} \in \mathcal{X}_{\text{caseS}}^N$ is ISS. ■

ACKNOWLEDGMENT

We thank Francesco Borrelli and Monimoy Bujarbaruah for valuable discussions.

REFERENCES

- [1] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, Madison, Wisconsin, 2 edition, 2017.
- [2] A. Bemporad and M. Morari. *Robust model predictive control: A survey*, pages 207–226. Springer London, London, 1999.
- [3] A. Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, 36(6):30–44, 2016.
- [4] M. Farina, L. Giulioni, and R. Scattolini. Stochastic linear model predictive control with chance constraints – a review. *Journal of Process Control*, 44:53–67, 2016.
- [5] T. Brüdigam, V. Gaßmann, D. Wollherr, and M. Leibold. Minimization of constraint violation probability in model predictive control. *International Journal of Robust and Nonlinear Control*, 31(14):6740–6772, 2021.
- [6] M. Fink, T. Brüdigam, D. Wollherr, and M. Leibold. Constraint violation probability minimization for norm-constrained linear model predictive control. In *2022 European Control Conference (ECC)*, pages 839–846, 2022.
- [7] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [8] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. Camacho. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1):5–21, 2009.
- [9] W. Langson, I. Chrysochoos, S. V. Raković, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.
- [10] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer. A computationally efficient robust model predictive control framework for uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 66(2):794–801, 2021.
- [11] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe. Safe and fast tracking on a robot manipulator: Robust mpc and neural network control. *IEEE Robotics and Automation Letters*, 5(2):3050–3057, 2020.
- [12] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823, 2014.
- [13] R. Soloperto, J. Köhler, F. Allgöwer, and M. A. Müller. Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control. In *2019 European Control Conference (ECC)*, pages 811–817, Naples, Italy, 2019.
- [14] L. Blackmore, M. Ono, A. Bektassov, and B.C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics*, 26(3):502–517, 2010.
- [15] 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*, 50(12):3009–3018, 2014.
- [16] M. C. Campi and S. Garatti. A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of optimization theory and applications*, 148(2):257–280, 2011.
- [17] T. Brüdigam, J. Teutsch, D. Wollherr, and M. Leibold. Combined robust and stochastic model predictive control for models of different granularity. *IFAC-PapersOnLine*, 53(2):7123–7129, 2020. 21st IFAC World Congress.
- [18] T. Brüdigam, J. Teutsch, D. Wollherr, M. Leibold, and M. Buss. Probabilistic model predictive control for extended prediction horizons. *at - Automatisierungstechnik*, 69(9):759–770, 2021.
- [19] L. Hewing, K. P. Wabersich, and M. N. Zeilinger. Recursively feasible stochastic model predictive control using indirect feedback. *Automatica*, 119:109095, 2020.
- [20] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer. Constraint-tightening and stability in stochastic model predictive control. *IEEE Transactions on Automatic Control*, 62(7):3165–3177, 2017.
- [21] F. Oldewurtel, C. N. Jones, A. Parisio, and M. Morari. Stochastic model predictive control for building climate control. *IEEE Transactions on Control Systems Technology*, 22(3):1198–1205, 2014.
- [22] Y. Ma, J. Matuško, and F. Borrelli. Stochastic model predictive control for building hvac systems: Complexity and conservatism. *IEEE Transactions on Control Systems Technology*, 23(1):101–116, 2015.
- [23] S. di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovskiy. Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management. *IEEE Transactions on Control Systems Technology*, 22(3):1018–1031, 2014.
- [24] X. Zeng and J. Wang. A parallel hybrid electric vehicle energy management strategy using stochastic model predictive control with road grade preview. *IEEE Transactions on Control Systems Technology*, 23(6):2416–2423, 2015.
- [25] G. Schildbach and F. Borrelli. Scenario model predictive control for lane change assistance on highways. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 611–616, Seoul, South Korea, 2015.
- [26] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli. Scenario model predictive control for lane change assistance and autonomous driving on highways. *IEEE Intelligent Transportation Systems Magazine*, 9(3):23–35, 2017.
- [27] T. Brüdigam, F. di Luzio, L. Pallottino, D. Wollherr, and M. Leibold. Grid-based stochastic model predictive control for trajectory planning in uncertain environments. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8, Rhodes, Greece, 2020.
- [28] T. Brüdigam, J. Zhan, D. Wollherr, and M. Leibold. Collision avoidance with stochastic model predictive control for systems with a twofold uncertainty structure. In *2021 IEEE 24th International Conference on Intelligent Transportation Systems (ITSC)*, pages 432–438, Indianapolis, IN, USA, 2021.
- [29] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold. Stochastic model predictive control with a safety guarantee for automated driving. *IEEE Transactions on Intelligent Vehicles*, pages 1–1, 2021.
- [30] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29(3):972–983, 2021.
- [31] M. Cannon, B. Kouvaritakis, and X. Wu. Probabilistic constrained mpc for multiplicative and additive stochastic uncertainty. *IEEE Transactions on Automatic Control*, 54(7):1626–1632, 2009.
- [32] M. Cannon, B. Kouvaritakis, and X. Wu. Model predictive control for systems with stochastic multiplicative uncertainty and probabilistic constraints. *Automatica*, 45(1):167–172, 2009.
- [33] W. Wojsznis, A. Mehta, P. Wojsznis, D. Thiele, and T. Blevins. Multi-objective optimization for model predictive control. *ISA Transactions*, 46(3):351–361, 2007.
- [34] A. Bemporad and D. Muñoz de la Peña. Multiobjective model predictive control. *Automatica*, 45(12):2823–2830, 2009.
- [35] F. Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.
- [36] Feng Xu. Minimal detectable and isolable faults of active fault diagnosis. *IEEE Transactions on Automatic Control*, 68(2):1138–1145, 2023.
- [37] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006.
- [38] R. H. G. Tan and L. Y. H. Hoo. DC-DC converter modeling and simulation using state space approach. In *2015 IEEE Conference on Energy Conversion (CENCON)*, pages 42–47, 2015.
- [39] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *2013 European control conference (ECC)*, pages 502–510, Zürich, Switzerland, 2013.
- [40] H. K. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002.
- [41] M. Bujarbaruah, X. Zhang, M. Tanaskovic, and F. Borrelli. Adaptive stochastic mpc under time-varying uncertainty. *IEEE Transactions on Automatic Control*, 66(6):2840–2845, 2021.