

Multi-Step Model Predictive Safety Filters: Reducing Chattering by Increasing the Prediction Horizon

Federico Pizarro Bejarano, Lukas Brunke, and Angela P. Schoellig

Abstract— Learning-based controllers have demonstrated superior performance compared to classical controllers in various tasks. However, providing safety guarantees is not trivial. Safety, the satisfaction of state and input constraints, can be guaranteed by augmenting the learned control policy with a safety filter. Model predictive safety filters (MPSFs) are a common safety filtering approach based on model predictive control (MPC). MPSFs seek to guarantee safety while minimizing the difference between the proposed and applied inputs in the immediate next time step. This limited foresight can lead to jerky motions and undesired oscillations close to constraint boundaries, known as chattering. In this paper, we reduce chattering by considering input corrections over a longer horizon. Under the assumption of bounded model uncertainties, we prove recursive feasibility using techniques from robust MPC. We verified the proposed approach in both extensive simulation and quadrotor experiments. In experiments with a Crazyflie 2.0 drone, we show that, in addition to preserving the desired safety guarantees, the proposed MPSF reduces chattering by more than a factor of 4 compared to previous MPSF formulations.

I. INTRODUCTION

Robotic systems are increasingly deployed to perform tasks in complex environments, including autonomous driving [1], aerial delivery [2], and surgery [3]. These tasks often suffer from significant uncertainties in the system dynamics, rendering classical controllers less effective [4]. Learning-based controllers have demonstrated improved performance on complex tasks [5]–[7]. However, reinforcement learning controllers typically lack safety guarantees and rarely enforce state constraints such as road lane boundaries for a self-driving car [4]. This prevents reinforcement learning controllers from being deployed on safety-critical systems despite promising results and demonstrations. Combinations of machine learning and model-based control (notably learning-based MPC) have become popular methods of leveraging the benefits of both approaches. However, these approaches do not have the adaptability across diverse systems and tasks of model-free reinforcement learning [4].

Safety filters allow arbitrary controllers to be implemented with safety guarantees, including deep learning controllers [4]. Model predictive safety filters (MPSFs) [8] are a category of safety filters that leverage model predictive

The authors are with the Learning Systems and Robotics Lab (www.learnsyslab.org), University of Toronto, Canada, and affiliated with the University of Toronto Robotics Institute and the Vector Institute for Artificial Intelligence in Toronto. Lukas Brunke and Angela P. Schoellig are also with the Technical University of Munich and the Munich Institute for Robotics and Machine Intelligence (MIRMI), Germany. E-mails: {federico.pizarrobejarano, lukas.brunke, angela.schoellig}@robotics.utoronto.ca

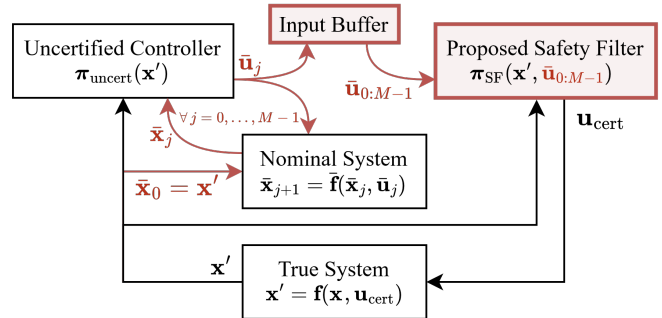


Fig. 1: Our proposed safety filter approach π_{SF} for safely controlling the system $\mathbf{x}' = \mathbf{f}(\mathbf{x}, \mathbf{u})$, with the novel components highlighted in red. The safety filter simulates an input trajectory for the next M time steps, using the nominal system \mathbf{f} and the uncertified controller $\pi_{uncert}(\mathbf{x}')$. The input trajectory $\bar{\mathbf{u}}_{0:M-1}$ along with the state \mathbf{x}' is passed to the safety filter π_{SF} , which outputs a safe control input \mathbf{u}_{cert} .

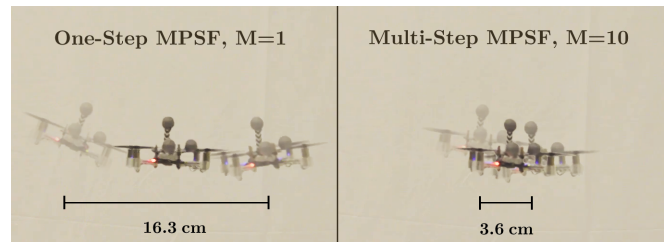


Fig. 2: Experimental results on a Crazyflie 2.0 drone illustrating the chattering caused by the standard one-step MPSF versus the proposed multi-step MPSF with a filtering horizon of $M = 10$. The proposed filter reduces the peak-to-peak amplitude of chattering near the constraint boundary from an average of 16.3 cm to 3.6 cm (78% reduction).

control (MPC) to predict whether uncertified (i.e., potentially unsafe) inputs sent from the controller will violate the constraints. In the case of a potential future constraint violation, the MPSF determines the minimal deviation from the uncertified input that results in constraint satisfaction (see Fig. 1).

Despite strong theoretical guarantees inherited from the underlying MPC [8], MPSFs may cause chattering and high-magnitude corrections to the uncertified control inputs (see Fig. 2). Chattering occurs when the controller directs the system towards a constraint boundary and is repeatedly stopped by the safety filter. This leads to an oscillatory behavior between the controller and the safety filter [9]. Additionally, due to the MPSF objective only looking ahead one time step, the filter may allow a system to approach the boundary very closely before using a high-magnitude correction to keep the system inside the state constraints. Both of these behaviors can cause the system to leave the operation regime as the high-frequency input changes may not be accounted for in

the system identification, increasing the model mismatch between the actual system and the nominal model in the MPSF [4], potentially leading to constraint violations.

Contributions: First, we propose generalizing the standard MPSF objective function to reduce chattering and other potentially unsafe corrective actions. Our approach inherits the theoretical recursive feasibility guarantees of the underlying MPC. Second, we propose metrics that reflect the magnitude of corrections and chattering. Previously, chattering has only been qualitatively described [8]–[10]. The efficacy of the proposed MPSF is demonstrated in simulation using the `safe-control-gym` [11] cartpole and quadrotor systems, and on real-world Crazyflie 2.0 quadrotors (see Fig. 2). As far as the authors know, this is only the second paper to implement MPSFs on real robots, where [12] designed an MPSF specific for miniature remote-controlled car racing. Our code can be found at <https://tinyurl.com/mpsf-code>, and a video of our experiment can be found at <https://tinyurl.com/mpsf-video>.

II. RELATED WORK

As learning-based controllers have grown in popularity, leveraging their strengths while also guaranteeing constraint satisfaction has become increasingly important. Many approaches for incorporating safety into model-free learning controllers serve to encourage safe behavior, reducing the probability of constraint violations. However, they do not provide hard guarantees on constraint satisfaction and are not generalizable to all learning controllers [4], [13].

Safety filters are necessary to guarantee the safety of an arbitrary controller. This is done by computing the minimal deviation from the latest uncertified control input that results in constraint satisfaction.

Control barrier functions (CBFs) are frequently used in safety filtering frameworks [4]. The CBFs define safe sets as their super-level set [14] and solve an optimization problem for each proposed control input to certify whether it is safe, and modify it otherwise. CBFs are usually defined for continuous-time systems [4], which poses challenges when running discrete-time controllers. Finding CBFs can be difficult and requires a model of the system dynamics or offline data; analytical solutions can typically only be found for special cases [14]. However, various methods for learning the CBF condition have recently been proposed [15]–[17]. Another approach to certification using a safe set determines the set using Hamilton-Jacobi reachability analysis [18] and switches between the original controller and a safe controller to keep the system inside the safe set [10]. The CBF- and reachability-based safety filters compute safe inputs quickly and have strong theoretical guarantees, but determining an appropriate safe set is challenging, and accounting for constraints that change during operation is typically computationally infeasible [4].

Model predictive safety filters (MPSFs) determine certified control inputs using model predictive control [8]. At each iteration, the MPSF receives a potentially unsafe input and determines the state of the system after this input is applied

using a model of the system dynamics. Then, from that future state, the MPSF uses the nominal model of the system to generate a trajectory to a safe terminal set in a number of steps up to the horizon of the MPC. This trajectory must be safe despite uncertainties, which is guaranteed using a known bound on the mismatch between the true system dynamics and the nominal model. If there is no feasible trajectory that will take the system from this future state to the terminal set, it will modify the original input so that this is possible.

MPSFs reduce offline computation and complex analytical work necessary for determining a safe set in favor of increased online computation [8]. Since MPSFs are based on well-established MPC frameworks, existing optimization techniques allow MPSFs to be executed in real-time [19], [20]. This makes them more suitable for complex systems in high dimensions, for which determining safety for all initial conditions is computationally infeasible, but for which safety can be guaranteed for a specific initial condition.

MPSFs, as well as other safety filters such as CBFs and reachability-based filters, can cause chattering [8], [9]. This has only been partially addressed for MPSFs by penalizing the rate of change of the inputs [12], a common MPC approach for achieving smooth state and input trajectories [21]. However, this approach does not directly penalize corrections but reduces the rate of change of the inputs over the entire trajectory, even during safe actions, and thus is not generally suitable. Reducing chattering is the main goal of this paper.

III. PROBLEM FORMULATION

We consider a discrete, time-invariant system

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad (1)$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state at time step k , $\mathbf{u}_k \in \mathbb{R}^m$ is the input, and \mathbf{f} describes the dynamic behavior. The system is subject to known state and input constraints, $\mathbf{x} \in \mathbb{X}_c$ and $\mathbf{u} \in \mathbb{U}_c$, where $\mathbb{X}_c \subset \mathbb{R}^n$ is closed and $\mathbb{U}_c \subset \mathbb{R}^m$ is compact.

We assume that we only have access to a nominal model $\bar{\mathbf{f}}$ and uncertainty set \mathbb{W} such that

$$\mathbf{x}_{k+1} = \bar{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}(\mathbf{x}_k, \mathbf{u}_k), \quad (2)$$

where $\mathbf{w}(\mathbf{x}, \mathbf{u}) \in \mathbb{W} \subset \mathbb{R}^n$, $\forall \mathbf{x} \in \mathbb{X}_c$, $\forall \mathbf{u} \in \mathbb{U}_c$, and \mathbb{W} is compact.

A control policy $\pi_{\text{uncert}}(\mathbf{x}_k)$, which may not enforce constraints, is used to achieve a task (e.g., trajectory tracking). A safety filter, which has access to the nominal model $\bar{\mathbf{f}}$, takes in the input $\mathbf{u}_{\text{uncert},k} = \pi_{\text{uncert}}(\mathbf{x}_k)$ as well as the current state of the system \mathbf{x}_k , and uses the nominal system to determine if the proposed input is safe (i.e., satisfies state and input constraints, and will not lead to a constraint violation in the future). If it is not safe, it will find a safe input $\mathbf{u}_{\text{cert},k}$ that minimally modifies the uncertified input. This certified input is the input applied to the system, $\mathbf{u}_k = \mathbf{u}_{\text{cert},k}$.

The goal of this paper is to address chattering, an undesirable behavior caused by safety filters [8], [9], seen in Fig. 2. We observe chattering when the input to the system oscillates between the original, unmodified input (because it is safe) and a corrected, certified input provided by the

safety filter (because the original input is unsafe). Consider a controller approaching a constraint boundary certified by a safety filter. Suppose the uncertified controller is going to violate the constraints. In that case, the safety filter will intervene, providing a safe input that may move the system in a direction opposite to the controller's intention. The controller may then reapproach the boundary in the next time steps, causing the safety filter to once again correct the input.

This oscillation deteriorates the performance and, due to the rapid input changes near the boundary, can lead to a constraint violation. This is because the safety guarantees of the safety filter depend on assumptions on the nominal model of the system and on the model error bounds, which may not be accurate for high-frequency input changes that have not been accounted for during system identification [4]. Additionally, safety filters may provide high-magnitude corrections, which can also lead to the deterioration of performance and safety.

IV. BACKGROUND

Although MPSFs can be applied to any underlying MPC framework, including output feedback MPC [22], we will use robust MPC with state feedback as the backbone of this paper. Robust MPC is a standard method of guaranteeing safety under uncertainty.

Definition 1 (Robust pos. control inv. (RPCI) set [21]):

A set $\mathbb{P} \subseteq \mathbb{X}_c$ is robust positively control invariant for the system in (2) with a controller $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ subject to state constraints \mathbb{X}_c , input constraints \mathbb{U}_c , and bounded uncertainty set \mathbb{W} , if the initial system state $\mathbf{x}_0 \in \mathbb{P}$ implies $\mathbf{x}_{k+1} = \bar{\mathbf{f}}(\mathbf{x}_k, \pi(\mathbf{x}_k)) + \mathbf{w}_k \in \mathbb{P}, \forall \mathbf{w}_k \in \mathbb{W}, \forall k \geq 0$.

Consider a robust tube-based MPC [23], [24] performing a task on the system \mathbf{f} . Let us denote the set of possible states of the system given the known uncertainty set at time step k as \mathbb{X}_k , and the evolution of the system as $\mathbb{X}_{k+1} \supseteq \Phi(\mathbb{X}_k, \mathbf{u}_k, \mathbb{W}) = \{\bar{\mathbf{f}}(\mathbf{x}, \mathbf{u}_k) + \mathbf{w} \in \mathbb{R}^n \mid \forall \mathbf{x} \in \mathbb{X}_k, \mathbf{w} \in \mathbb{W}\}$.

Assumption 1: There exists a terminal set $\mathbb{X}_{\text{term}} \subset \mathbb{X}_c$ and a terminal controller $\pi_{\text{term}} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that the following properties hold for all $\mathbf{x} \in \mathbb{X}_{\text{term}}$:

$$\begin{aligned} \Phi(\mathbb{X}_{\text{term}}, \pi_{\text{term}}(\mathbf{x}), \mathbb{W}) &\subseteq \mathbb{X}_{\text{term}} \\ \pi_{\text{term}}(\mathbf{x}) &\in \mathbb{U}_c. \end{aligned}$$

Thus, \mathbb{X}_{term} is an RCPI set under the terminal controller π_{term} .

Generally, MPC solves an optimization problem at each time step for an optimal input sequence over the next H time steps, where $H \in \mathbb{N}$ is the horizon. At each time step k , MPC only applies the first input from the resulting optimal input sequence. The optimization problem for a general robust, tube-based MPC can be stated as follows:

$$\min_{\mathbf{u}_{0|k}, \dots, \mathbf{u}_{H-1|k}} \sum_{j=0}^{H-1} \ell(\mathbb{X}_{j|k}, \mathbf{u}_{j|k}) + \ell_{\text{term}}(\mathbb{X}_{H|k}) \quad (3a)$$

$$\text{s.t. } \mathbf{x}_k \in \mathbb{X}_{0|k} \quad (3b)$$

$$\mathbb{X}_{i+1|k} \supseteq \Phi(\mathbb{X}_{i|k}, \mathbf{u}_{i|k}, \mathbb{W}) \quad (3c)$$

$$\mathbb{X}_{i|k} \subseteq \mathbb{X}_c, \forall i = 0, \dots, H-1 \quad (3d)$$

$$\mathbf{u}_{i|k} \in \mathbb{U}_c \quad (3e)$$

$$\mathbb{X}_{H|k} \subset \mathbb{X}_{\text{term}}, \quad (3f)$$

where $\ell(\cdot)$ and $\ell_{\text{term}}(\cdot)$ are the stage and terminal cost functions, respectively, $\mathbb{X}_{i|k}$ is the set of possible states at the $(k+i)$ -th time step computed at time step k , and $\mathbf{u}_{i|k}$ is the input at the $(k+i)$ -th time step computed at time step k .

MPSFs certify a proposed input $\mathbf{u}_{\text{uncert},k}$ by replacing the MPC objective function (3a) with

$$J_{\text{MPSF},1}(\mathbf{x}_k) = \|\pi_{\text{uncert}}(\mathbf{x}_k) - \mathbf{u}_{0|k}\|_{\mathbf{R}}, \quad (4)$$

where $\mathbf{R} \in \mathbb{R}^{m \times m}$ is a positive semi-definite cost matrix to weigh different input components. This objective function minimizes the difference between the next safe input $\mathbf{u}_{0|k}$ and the incoming uncertified input $\mathbf{u}_{\text{uncert},k}$. Therefore, the MPSF is a one-step input reference tracking MPC that tracks the input reference provided by an uncertified controller. Note that the MPSF inherits the recursive feasibility guarantees from the underlying MPC formulation with no additional assumptions required.

We will denote this standard MPSF objective function as the 'one-step' objective function, as it minimizes the difference between the certified and uncertified inputs only for the upcoming time step. The remainder of the paper will deal with augmenting this objective function to improve performance and safety.

In [12] a term is added to the objective in (4) to directly penalize changes in the input. This is a common MPC approach for achieving smooth state and input trajectories [21]. The additional penalization term is defined as follows:

$$J_{\text{reg},M_r} = \sum_{j=0}^{M_r-1} w_r(j) \|\Delta \mathbf{u}_{j|k}\|_{\mathbf{R}_r}^2, \quad (5)$$

where $w_r(\cdot) : \mathbb{N}_0 \rightarrow \mathbb{R}^+$ calculates the weight associated with the j -th input rate, M_r is the regularization horizon, $\Delta \mathbf{u}_{j|k} = \mathbf{u}_{j|k} - \mathbf{u}_{j-1|k}$, $\Delta \mathbf{u}_{0|k} = \mathbf{u}_{0|k} - \mathbf{u}_{0|k-1}$, and $\mathbf{R}_r \in \mathbb{R}^{m \times m}$ is a cost matrix for the input rates.

This approach directly penalizes the rate of change of the inputs, and thus it will correct the controller even when the uncertified inputs are safe, which is not the purpose of safety filters. It was used only in [12], which defines an MPSF specifically for controlling a remote-controlled miniature car for racing through a fixed-width track. This additional term was used due to the necessity of smooth trajectories in that specific experiment, but the term is not generally applicable. If the controller was executing high-frequency commands without violating constraints, a standard safety filter (including our proposed approach) would not interfere while the regularized approach would unnecessarily modify the control inputs to reduce the rate of change of the inputs. As the controller should not be interrupted when the constraints are inactive, such as to leverage the performance and flexibility of learning-based controllers, this regularization approach is unsuitable for most safety filtering tasks. See Figure 8 for an illustration of this undesired behavior.

Another approach would be to constrain on how much the input can change, which may reflect fundamental limitations of the system or simply be used to reduce the maximum input change. However, this maximum input change would have to

be carefully tuned, and would potentially limit the flexibility of the system, and the constraint would not affect the system if the input changes are smaller than this maximum value.

V. MULTI-STEP MODEL PREDICTIVE SAFETY FILTERS

In this section, we propose our novel objective function that leverages the nominal system dynamics and predictions of the behavior of the controller.

A. Filtering Objectives with Increased Horizon Lengths

The standard one-step MPSF objective function (4) minimizes the difference between the current uncertified input and the immediate certified input. However, it does not reduce future corrections and thus allows chattering and jerky corrections. To reduce this, we propose additionally including upcoming corrections in the MPSF objective function:

$$J_{\text{MPSF},M}(\mathbf{x}_k) = \sum_{j=0}^{M-1} w(j) \|\pi_{\text{uncert}}(\mathbf{x}_{j|k}) - \mathbf{u}_{j|k}\|_{\mathbf{R}}^2, \quad (6)$$

where $w(\cdot) : \mathbb{N}_0 \rightarrow \mathbb{R}^+$ calculates the weights associated with the j -th correction and M is the filtering horizon.

In practice, the weight $w(\cdot)$ and the filtering horizon M are tuning parameters. The weight $w(\cdot)$ controls the impact of future proposed inputs. A constant $w(\cdot)$ may be used when all future proposed control inputs are equally important, while a decreasing $w(\cdot)$ may be applied when future corrections should have less impact as j increases. The filtering horizon M determines how many time steps into the future the safety filter considers when minimizing the corrections. Intuitively, larger M values cause the safety filter to intervene early and proactively. Note that our proposed formulation generalizes the one-step MPSF, which is recovered by setting $M = 1$.

The multi-step MPSF optimization problem is then

$$\min_{\mathbf{u}_{0|k}, \dots, \mathbf{u}_{H-1|k}} \sum_{j=0}^{M-1} w(j) \|\pi_{\text{uncert}}(\mathbf{x}_{j|k}) - \mathbf{u}_{j|k}\|_{\mathbf{R}}^2 \quad (7a)$$

$$\text{s.t. constraints (3b) – (3f)} \quad (7b)$$

The following theorem summarizes the proposed MPSF's recursive feasibility guarantees inherited from the underlying MPC formulation.

Theorem 1: Consider the system to be controlled as in (1) and let (2) be satisfied with a nominal model $\bar{\mathbf{f}}$ and a compact uncertainty set \mathbb{W} . Furthermore, there exist a terminal constraint set \mathbb{X}_{term} and an associated terminal controller π_{term} according to Assumption 1. Let the optimization in (7) be feasible at time step $k = 0$. Then the optimization in (7) is feasible for all $k \in \mathbb{N}$.

Proof: The optimal input trajectory at time step $k = 0$, $\{\mathbf{u}_{0|0}^*, \mathbf{u}_{1|0}^*, \dots, \mathbf{u}_{H-1|0}^*\}$, is feasible by assumption. Then, the system will be within the terminal set by time step H , since the feasibility of the terminal constraint yields $\mathbb{X}_{H|0} \subset \mathbb{X}_{\text{term}}$. After applying $\mathbf{u}_{0|0}$, the system evolves to $\mathbb{X}_1 \supseteq \Phi(\mathbb{X}_0, \mathbf{u}_{0|0}, \mathbb{W})$. Then initializing (7) at $k = 1$ with \mathbb{X}_1 , we know that the input sequence $\{\mathbf{u}_{1|0}, \dots, \mathbf{u}_{H-1|0}, \pi_{\text{term}}(\mathbf{x}_{H|0})\}$ is feasible for all $\mathbf{x}_{H|0} \in \mathbb{X}_{H|0} \subset \mathbb{X}_{\text{term}}$. This yields $\mathbb{X}_{H|1} \subset \mathbb{X}_{\text{term}}$ by Assumption 1. This can be repeated for

all time steps $k \in \mathbb{N}$ and ensures there exists a feasible input trajectory that keeps the system within \mathbb{X}_{term} at time step $k + H$, and thus within the constraints \mathbb{X}_{c} . ■

B. Approximate Black Box Control Policies

The controller $\pi_{\text{uncert}}(\mathbf{x}_k)$ may be a black box or a non-linear controller such as a reinforcement learning control policy represented by a deep neural network. Thus, the closed-loop behavior of the controller and safety filter must be approximated in (6). Although many approximations are possible, we found that simulating the behavior of the uncertified controller using the nominal model was the most effective approach and requires no additional learning or tuning beyond M and $w(\cdot)$.

We approximate the closed-loop behavior of the uncertified controller by simulating the effect of the uncertified controller on the nominal model (see Fig. 1). At each time step k , the simulated trajectory is calculated for the next M steps as follows:

$$\begin{aligned} \bar{\mathbf{u}}_{\text{uncert},j|k} &= \pi_{\text{uncert}}(\bar{\mathbf{x}}_{j|k}) \\ \bar{\mathbf{x}}_{j+1|k} &= \bar{\mathbf{f}}(\bar{\mathbf{x}}_{j|k}, \bar{\mathbf{u}}_{\text{uncert},j|k}), \forall j = 0, \dots, M-1, \end{aligned}$$

where $\bar{\mathbf{x}}_{j|k}$ represents the j -th state of the simulated state trajectory calculated at time step k and, analogously, $\bar{\mathbf{u}}_{\text{uncert},j|k}$ is the simulated input trajectory. We set $\bar{\mathbf{x}}_{0|k} = \mathbf{x}_k$. Thus, the controller is approximated as $\pi_{\text{uncert}}(\mathbf{x}_{k+j}) \approx \bar{\mathbf{u}}_{\text{uncert},j|k}$. The weight function $w(j)$ can be chosen to reflect the accuracy of this approximation.

This approach requires no tuning but does necessitate access to the controller for the simulation. Thus, this approach is not suitable for human teleoperation, since the controller cannot be queried, but is appropriate for black box controllers or reinforcement learning controllers, which are difficult to approximate but can be simulated. If the controller is inaccessible, other approximations of the controller can be used. Various other approaches, such as approximating the controller as a linear quadratic regulator (LQR) or using linear regression to learn a local approximation of the control policy, were tested by the authors of this paper and found to have comparable results to the proposed approximation. However, we restrict the presentation to the simulated input trajectory due to its simplicity, lack of additional tuning parameters, and overall efficacy.

VI. SAFETY FILTER PERFORMANCE METRICS

Previous safety filter research has focused on eliminating constraint violations of the uncertified controller. In addition, this paper focuses on minimizing the undesirable effects of safety filters such as unnecessary corrections, jerkiness, and chattering. We propose several new metrics to measure the interventions of the safety filter and the chattering.

We consider a matrix of corrections, $\mathbf{C} \in \mathbb{R}^{K \times m}$, where K is the total number of time steps in the experiment. Each row of \mathbf{C} , $\mathbf{c}_k = \mathbf{R}^{1/2}(\mathbf{u}_{\text{uncert},k} - \mathbf{u}_{\text{cert},k}) \in \mathbb{R}^m$, is the correction at time step $k = 0, \dots, K-1$, and $\mathbf{R}^{1/2}$ is the positive semi-definite square-root matrix of \mathbf{R} . Thus, the safety filter at time step k minimizes the norm of the correction $\|\mathbf{c}_k\|_2^2$. The cost

matrix \mathbf{R} in (4) and (6) weighs the different components of the input. For example, it can be chosen as the identity if the inputs are normalized and of the same size. However, if \mathbf{R} is not identity, we account for it in the corrections by weighing them using its square-root matrix.

A. Measuring Interventions

A safety filter should minimize how significantly and how often it must intervene. We propose to measure this in two ways: the magnitude of the corrections and the number of corrections.

The magnitude of corrections is calculated as the norm of the matrix of corrections, $\|\mathbf{C}\|_F$, where $\|\cdot\|_F$ is the Frobenius norm. This measures how much the safety filter modified the proposed inputs in an experiment. The Frobenius norm is used as it weighs all elements of a matrix equally, extending the Euclidean norm to matrices.

The number of corrections indicates the number of time steps at which the proposed input was modified above a certain percentage tolerance ϵ . This is calculated as $\sum_{k=0}^{K-1} \mathbf{1}\left(\frac{\|\mathbf{c}_k\|_2}{\|\mathbf{u}_{\text{cert},k}\|_R} \geq \epsilon\right)$, where $\mathbf{1}(\cdot)$ is the indicator function. This equation simply counts the times when the magnitude of the correction divided by the magnitude of the certified input is above the tolerance ϵ . The tolerance will depend on how sensitive the system is to variations in the input.

B. Measuring Chattering

Safety filters usually increase the rate of change of the inputs through chattering and high-magnitude corrections. To measure a safety filter's impact, we propose two metrics: the maximum correction and the norm of the rate of change of the inputs.

The maximum correction is $\max_{k \in \{0, \dots, K-1\}} \|\mathbf{c}_k\|_2$. Higher maximum corrections typically correspond to jerky movement. Ideally, we wish to maintain safety with small-magnitude corrections.

The rate of change of the inputs measures how much the input varied during an experiment, and is significantly increased by chattering and jerky inputs. Consider the applied inputs \mathbf{u}_k for $k = 0, \dots, K-1$. First, we take the numerical derivative $\delta \mathbf{u}_k = \mathbf{R}^{1/2}(\mathbf{u}_k - \mathbf{u}_{k-1})/\delta t$ for $k = 1, \dots, K-1$ (where δt is the length of a time step) and stack them into a matrix $\Delta \mathbf{u} = [\delta \mathbf{u}_1, \dots, \delta \mathbf{u}_{K-1}]$. Then the norm of the rate of change of the inputs is $\|\Delta \mathbf{u}\|_F$. This metric can also be applied to an uncertified trajectory, enabling comparisons between uncertified and filtered controllers.

VII. EXPERIMENTAL RESULTS

To determine the efficacy of the proposed multi-step MPSF, we ran experiments in the safe learning-based control simulation environment `safe-control-gym` [11] and on a real quadrotor, the Crazyflie 2.0 (see Fig. 2 and Fig. 9). The underlying MPC is a robust nonlinear MPC formulation [24], which assumes time-invariant system dynamics. Additionally, it assumes that the system is incrementally stabilizable and the model mismatch between the nominal system and the real system is bounded by a known bound. The upper

bound on the model mismatch was found experimentally by comparing the true states and the states predicted by the nominal model while executing trajectories for system identification, $w_{\max} = \max_{k \in \{0, \dots, K-1\}} \|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{1|k}\|_2$, and thus we define $\mathbb{W} = \{\mathbf{w} \in \mathbb{R}^n \mid \|\mathbf{w}\|_2 \leq w_{\max}\}$.

The experiments test the performance of the standard one-step MPSF compared to our proposed multi-step MPSF with $M = 2, 5, 10$. Additionally, we consider the one-step MPSF with regularization (5) (with $M_r = 10$) to demonstrate our approach is effective even compared to an approach that directly penalizes the rate of change of the inputs along the entire trajectory, which leads to over-correcting that is unsuitable for most safety filtering tasks. For all experiments, the weight functions $w(j)$ and $w_r(j)$ were chosen to be $w(j) = w_r(j) = 0.85^j$ and the input channel cost matrices \mathbf{R} , and \mathbf{R}_r were chosen to be the identity matrix.

A. Simulation

The MPSFs were evaluated in simulation using the `safe-control-gym` [11]. The experiments were done on three different systems and two different tasks: a cartpole, a two-dimensional quadrotor, and a three-dimensional quadrotor, on both stabilization and trajectory tracking tasks with box constraints on the state and input. Because of space constraints, only the results of the cartpole trajectory tracking experiments are presented here. However, the experiments on the other systems and tasks led to similar conclusions. The remaining results can be found as part of our code release, see <https://tinyurl.com/mpsf-results>.

The cartpole system is a cart on a track with a pole hinged to the top of the cart. The state is $\mathbf{x} = [x, \dot{x}, \theta, \dot{\theta}]^T$, where x is the horizontal position of the cart, \dot{x} is the velocity of the cart, θ is the angle of the pole with respect to the vertical axis, and $\dot{\theta}$ is the angular velocity of the pole. The input $u \in \mathbb{R}$ is the force applied to the center of mass of the cart. There is no friction. The nominal equations of motion are [11]:

$$\begin{aligned} \ddot{x} &= \frac{u + m_p l (\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta)}{m_c + m_p} \\ \ddot{\theta} &= \frac{g \sin \theta + \cos \theta \left(\frac{-u - m_p l \dot{\theta}^2 \sin \theta}{m_c + m_p} \right)}{l \left(\frac{4}{3} - \frac{m_p \cos^2 \theta}{m_c + m_p} \right)}, \end{aligned}$$

where g is the gravitational acceleration, m_c and m_p are the masses of the cart and the pole, respectively, and l is half the length of the pole. The true behavior of the cartpole in the simulation is determined not by these equations but by the PyBullet [25] physics engine, which these equations seek to represent. These equations were discretized using a sampling time of $\delta t = \frac{1}{15}$ s to get the nominal discrete-time model of the cartpole, which was found to be very accurate, with $w_{\max} = 0.0014$. An accurate nominal model was chosen to demonstrate that MPSFs with $M = 1$ still cause chattering even when the model is accurately known. When the model is less accurately known chattering will still occur but the safety filter will be more conservative.

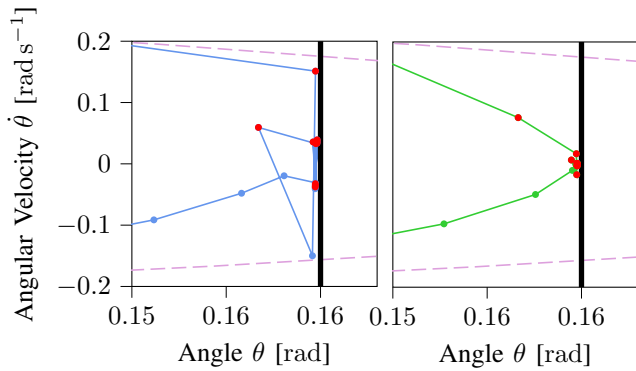


Fig. 3: Recorded trajectories of a cartpole trajectory tracking example when approaching a state constraint (in black) with the uncertified trajectory in purple (dashed), the one-step MPSF in blue (left), and the proposed MPSF with $M = 2$ in green (right). States at which the input was modified are indicated by a red dot. The standard one-step approach causes significant chattering, while the proposed approach greatly reduces the amplitude of the chattering.

The trajectory tracking task consists of tracking a sinusoidal reference in the x -direction with an amplitude of 1 m and a period of 5 s while constraining the angle of the pole to be within $[-9.2^\circ, 9.2^\circ]$ and the applied force to be within $[-10 \text{ N}, 10 \text{ N}]$. The chosen controllers are a linear-quadratic regulator (LQR), a proximal policy optimization (PPO) [26] reinforcement learning controller, and a soft actor-critic (SAC) [27] reinforcement learning controller. Each MPSF was tested with an MPC constraint horizon of $H = 20$ and a controller frequency of 15 Hz. To find suitable and interesting starting points, random states from the constrained state space were sampled. If a starting state caused at least five constraint violations when the controllers were uncertified, but could achieve constraint satisfaction in closed-loop with the one-step MPSF, then that starting point was selected. Ten of these starting points were generated for each uncertified controller. Each MPSF was tested using these ten starting states. The starting states are the source of randomness in the results.

In the experiments, chattering is significantly reduced by our proposed multi-step MPSF. In Fig. 3 we see an example of chattering when certified by a one-step MPSF versus when using the proposed MPSF. In cartpole trajectory tracking, the norm of the rate of change of the inputs compared to the one-step MPSF is reduced by up to 73% (see Fig. 4), the magnitude of corrections are reduced by up to 25% (see Fig. 5), and the maximum correction is reduced by up to 52% (see Fig. 6). This demonstrates that the proposed filter effectively reduces chattering, achieving a similar norm of the rate of change of the inputs compared to the uncertified control inputs (those, however, cause constraint violations), and generally decreases the overall correction effort as well.

The proposed filter has a comparable norm of the rate of change of inputs to the one-step MPSF with the additional regularization term but has up to a 22% lower magnitude of corrections and 29% lower maximum correction. Additionally, our approach does not correct the system when the constraints are inactive, reducing the number of corrections

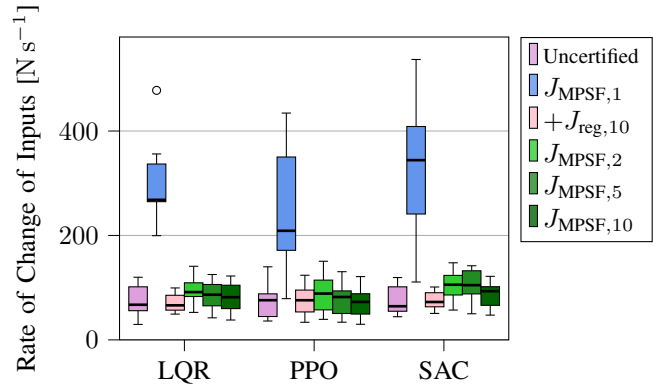


Fig. 4: The norm of the rate of change of the inputs (see Section VI-B) for the simulated cartpole trajectory tracking experiments for the uncertified controller, the multi-step MPSFs with varying M (denoted $J_{\text{MPSF},M}$), and a one-step MPSF with regularization (denoted $+J_{\text{reg},10}$). The proposed approach significantly decreases the norm of the rate of change of the inputs, up to a 73% decrease compared to the one-step approach (in blue), without violating the constraints.

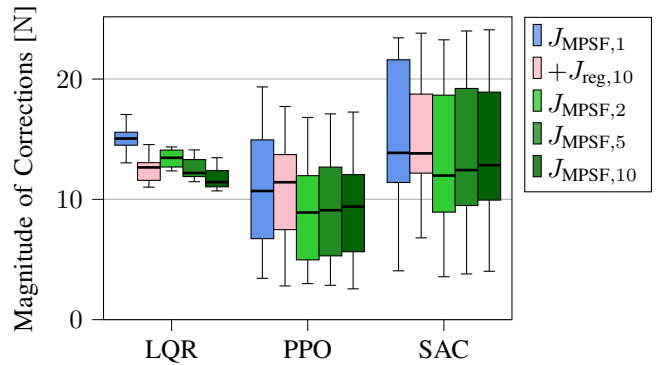


Fig. 5: The magnitude of corrections (see Section VI-A) for the simulated cartpole trajectory tracking experiments for the multi-step MPSFs with varying M (denoted $J_{\text{MPSF},M}$), and a one-step MPSF with regularization (denoted $+J_{\text{reg},10}$). The proposed approach decreases the median magnitude of corrections by up to 25% compared to the one-step approach, demonstrating it has an overall lower correction effort.

compared to the regularized approach by up to 66% (see Fig. 7). Higher M values do increase the number of corrections as the filter is intervening more proactively and early by considering corrections further in the future. However, the MPSF with $M = 10$ still has fewer interventions than the regularized approach. When tracking a safe but high-frequency trajectory (see Fig. 8), the one-step and multi-step MPSFs do not correct the controller as safety is not violated. However, the regularized MPSF significantly corrects the controller, unnecessarily deviating from the desired safe trajectory. This highlights that the additional regularization term is effective close to the safety boundary reducing chattering and high-magnitude corrections; however, it limits performance away from the safety boundary.

To address the computational aspects of our proposed approach, we observe an increase in the time required to solve the multi-step MPSF problem compared to the one-step MPSF. The time required to solve the MPSF at one time instance increased by approximately 7%, 22%, and 60% when $M = 2, 5, 10$, respectively. Adding regularization increased the execution time by 45%.

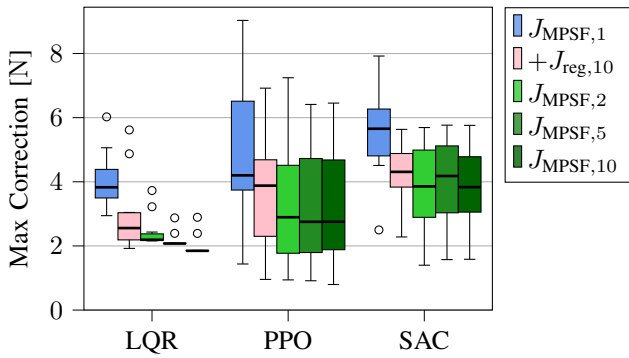


Fig. 6: The maximum correction (see Section VI-B) for the simulated cartpole trajectory tracking experiments for the multi-step MPSFs with varying M (denoted $J_{\text{MPSF},M}$), and a one-step MPSF with regularization (denoted $+J_{\text{reg},10}$). The proposed approach decreases the median maximum correction by up to 52% compared to the one-step approach, demonstrating it causes less jerky corrections.

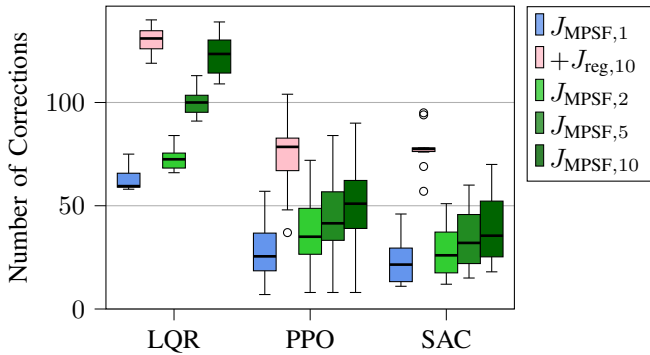


Fig. 7: The number of corrections (see Section VI-A) for the simulated cartpole trajectory tracking experiments, using a tolerance of $\epsilon = 0.1$, for the multi-step MPSFs with varying M (denoted $J_{\text{MPSF},M}$), and a one-step MPSF with regularization (denoted $+J_{\text{reg},10}$). Both the proposed approach and regularization increase the number of corrections compared to the one-step MPSF (in blue), but the proposed approach decreases the median number of corrections by up to 66% compared to the one-step MPSF with regularization.

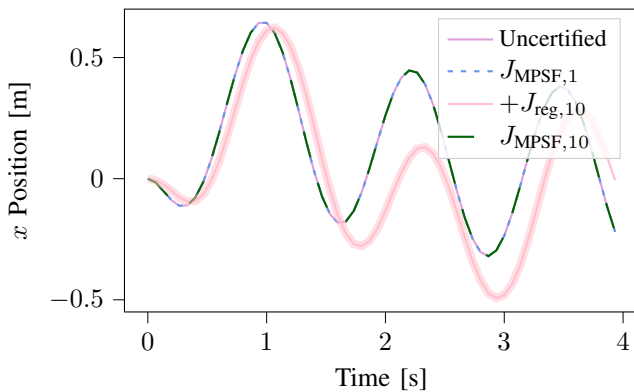


Fig. 8: Position trajectories of a simulated cartpole with a safe yet high-frequency sinusoidal reference trajectory when uncertified and when certified by the one-step ($J_{\text{MPSF},1}$), one-step with regularization ($+J_{\text{reg},10}$), and multi-step MPSFs ($J_{\text{MPSF},M}$). Time steps where the safety filter executed a correction (with a tolerance $\epsilon = 0.1$) are highlighted. Note that the one-step MPSF and the proposed MPSF do not correct the LQR controller as the trajectory is safe; however, the regularized approach does, causing a significant deviation from the desired trajectory.

B. Robot Experiments

The MPSFs were tested on a horizontal trajectory tracking task using a Crazyflie 2.0 quadrotor. The desired trajectory is a sinusoid in x with an amplitude of 1.5 m and a period of 10 s. The x position of the drone is constrained to be within $[-0.75 \text{ m}, 0.75 \text{ m}]$ and the velocity of the drone is constrained to be within $[-0.5 \text{ m s}^{-1}, 0.5 \text{ m s}^{-1}]$ (see Fig. 9). The input $u \in \mathbb{R}$ is a position setpoint the quadrotor tracks using an internal cascaded nonlinear controller [28]. This commanded position setpoint is constrained to be within 0.25 m of the current position to ensure smooth behavior from the internal position controller. The closed-loop dynamics in the x -direction can be approximated as a linear system. The linear system is identified by applying a sinusoidal reference trajectory with varying frequencies and amplitudes. We consider the state $\mathbf{x} = [x_1, x_2]^\top \in \mathbb{R}^2$, where x_1 is the quadrotor's position and x_2 is its velocity in the x -direction. Using MATLAB's System Identification Toolbox, we identified the system at 25 Hz, as

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.9756 & 0.0287 \\ -0.2793 & 0.8535 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0.0231 \\ 0.2854 \end{bmatrix} u_k + w_k,$$

which we use as the nominal model in the MPSF. The model mismatch was found to be significantly higher than for the simulation experiments, with $w_{\text{max}} = 0.0449$. The chosen controller is an LQR that causes violations in the position and velocity constraints. A motion capture system provides the quadrotor position.

The one-step, one-step with regularization (with $M_r = 10$), and our proposed multi-step (with $M = 2, 5, 10$) MPSFs are tested five times each with an MPC horizon of $H = 10$. Every test starts at the same initial position, the origin (and a static height of 1 m). As seen in Table I, in the experiments, our proposed approach significantly reduces the norm of the rate of change of the inputs, reducing it by 80% compared to the one-step approach when $M = 10$. The maximum correction and magnitude of corrections are either maintained or decreased, and both are decreased by over 30% compared to the one-step approach when $M = 10$. We see that the one-step with regularization is outperformed in every metric by the proposed approach with $M = 5$ and $M = 10$, including the norm of the rate of change of the inputs. This demonstrates that our proposed approach is more effective at decreasing chattering. This is achieved by considering (or predicting) chattering in the future as opposed to directly penalizing the rate of change of the inputs, while still decreasing the maximum correction and the magnitude of corrections.

VIII. CONCLUSION

This paper proposes a modified objective function for model predictive safety filters (MPSFs) that reduces chattering while not affecting performance when away from the safety boundaries. The paper also introduces several metrics to measure the performance of safety filters. The proposed approach minimizes the input corrections of the safety filter over a multi-step horizon rather than a single time step. The

TABLE I: Experimental results summary for 5 trials certifying an LQR controller flying a Crazyflie 2.0 on a sinusoidal path. The number of corrections uses a tolerance of $\epsilon = 0.1$ and is out of a total of 497 time steps.

Metric	Uncertified	$J_{\text{MPSF},1}$	$+J_{\text{reg},10}$	$J_{\text{MPSF},2}$	$J_{\text{MPSF},5}$	$J_{\text{MPSF},10}$
Mean norm of the rate of change of the inputs [m s^{-1}]	2.59	30.26	15.86	23.21	13.69	6.18
Mean magnitude of corrections [m]	-	2.96	2.67	2.89	2.51	2.03
Mean maximum correction [m]	-	0.46	0.44	0.45	0.42	0.30
Mean Number of Corrections	-	76.00	299.40	81.00	175.40	194.20

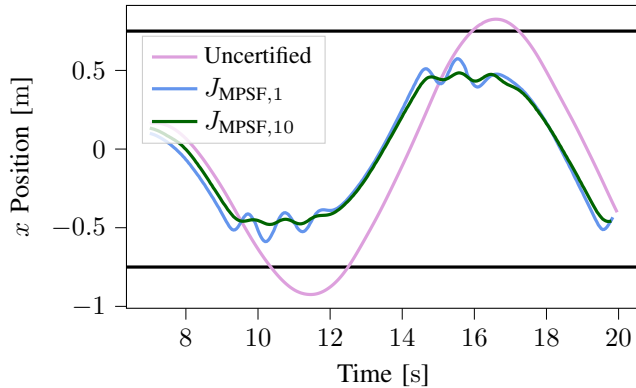


Fig. 9: Position trajectories of the real Crazyflie 2.0 experiment when uncertified, certified by the one-step MPSF ($J_{\text{MPSF},1}$), and certified by the proposed multi-step MPSF with $M = 10$ ($J_{\text{MPSF},10}$). The proposed filter reduces the peak-to-peak amplitude of chattering near the constraint boundaries (thick black lines), reducing it from an average of 16.3 cm to 3.6 cm. The certified controllers do not approach the constraint boundaries closely due to the large model uncertainty and strict velocity constraints.

future behavior of the controller is approximated by simulating its input trajectory leveraging a known nominal dynamics model. The presented approach requires little tuning and no additional assumptions compared to previous MPSFs. In simulation and real experiments, we found that the proposed multi-step MPSF greatly reduces chattering compared to the standard MPSF formulation, expanding the applicability of MPSFs to real-world systems.

REFERENCES

- [1] K. Burnett *et al.*, “Zeus: A system description of the two-time winner of the collegiate SAE AutoDrive competition,” *Journal of Field Robotics*, vol. 38, no. 1, 2021.
- [2] J. J. Boutilier *et al.*, “Optimizing a drone network to deliver automated external defibrillators,” *Circulation*, vol. 135, no. 25, 2017.
- [3] A. A. Shvets, A. Rakhlin, A. A. Kalinin, and V. I. Iglovikov, “Automatic instrument segmentation in robot-assisted surgery using deep learning,” in *IEEE International Conf. on Machine Learning and Applications (ICMLA)*, 2018.
- [4] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, “Safe learning in robotics: From learning-based control to safe reinforcement learning,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, 2022.
- [5] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, 2015.
- [6] D. Silver *et al.*, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, 2016.
- [7] OpenAI, I. Akkaya, *et al.*, “Solving rubik’s cube with a robot hand,” 2019.
- [8] K. P. Wabersich and M. N. Zeilinger, “Linear model predictive safety certification for learning-based control,” in *IEEE Conf. on Decision and Control (CDC)*, 2018.
- [9] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, “Learning-based model predictive control for safe exploration,” in *IEEE Conf. on Decision and Control (CDC)*, 2018.
- [10] A. K. Akametalu, S. Kaynama, J. F. Fisac, M. Zeilinger, J. Gillula, and C. Tomlin, “Reachability-based safe learning with Gaussian processes,” in *IEEE Conf. on Decision and Control*, 2014.
- [11] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig, “Safe-control-gym: A unified benchmark suite for safe learning-based control and reinforcement learning in robotics,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, 2022.
- [12] B. Tearle, K. P. Wabersich, A. Carron, and M. N. Zeilinger, “A predictive safety filter for learning-based racing control,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, 2021.
- [13] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, “Robust constrained learning-based NMPC enabling reliable mobile robot path tracking,” *The International Journal of Robotics Research*, vol. 35, no. 13, 2016.
- [14] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *European Control Conf. (ECC)*, 2019.
- [15] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, “Learning control barrier functions from expert demonstrations,” in *IEEE Conf. on Decision and Control (CDC)*, 2020.
- [16] L. Brunke, S. Zhou, and A. P. Schoellig, “Barrier bayesian linear regression: Online learning of control barrier conditions for safety-critical control of uncertain systems,” in *Proc. of The 4th Annual Learning for Dynamics and Control Conf.*, ser. Proceedings of Machine Learning Research, vol. 168, 2022.
- [17] Z. Qin, D. Sun, and C. Fan, “Sablas: Learning safe control for black-box dynamical systems,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, 2022.
- [18] I. Mitchell, A. Bayen, and C. Tomlin, “A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Transactions on Automatic Control*, vol. 50, no. 7, 2005.
- [19] Y. Wang and S. P. Boyd, “Fast model predictive control using online optimization,” *IEEE Transactions on Control Systems Technology*, vol. 18, 2010.
- [20] D. Kouzoupis, G. Frison, A. Zanelli, and M. Diehl, “Recent advances in quadratic programming algorithms for nonlinear model predictive control,” *Vietnam Journal of Mathematics*, vol. 46, 2018.
- [21] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*, 2nd ed. Madison, Wisconsin: Nob Hill Publishing, 2017.
- [22] L. Brunke, S. Zhou, and A. P. Schoellig, “Robust predictive output-feedback safety filter for uncertain nonlinear control systems,” in *IEEE Conf. on Decision and Control (CDC)*, 2022.
- [23] D. Q. Mayne, M. M. Seron, and S. Raković, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, no. 2, 2005.
- [24] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, “A computationally efficient robust model predictive control framework for uncertain nonlinear systems – extended version,” *IEEE Transactions on Automatic Control*, vol. 66, no. 2, 2021.
- [25] E. Coumans and Y. Bai, “PyBullet, a Python module for physics simulation for games, robotics and machine learning,” <http://pybullet.org>, 2016–2021.
- [26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv:1707.06347 [cs]*, 2017, arXiv: 1707.06347.
- [27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International Conference on Machine Learning*, 2018.
- [28] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *IEEE International Conf. on Robotics and Automation*, 2011.