2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Safety Filter Design for Neural Network Systems via Convex Optimization

Shaoru Chen†, Kong Yao Chee†, Nikolai Matni, M. Ani Hsieh, George J. Pappas

Abstract— **With the increase in data availability, it has been widely demonstrated that neural networks (NN) can capture complex system dynamics precisely in a data-driven manner. However, the architectural complexity and nonlinearity of the NNs make it challenging to synthesize a provably safe controller. In this work, we propose a novel safety filter that relies on convex optimization to ensure safety for a NN system, subject to additive disturbances that are capable of capturing modeling errors. Our approach leverages tools from NN verification to over-approximate NN dynamics with a set of linear bounds, followed by an application of robust linear MPC to search for controllers that can guarantee robust constraint satisfaction. We demonstrate the efficacy of the proposed framework numerically on a nonlinear pendulum system.**

I. Introduction

With the rapid development in machine learning infrastructure, neural networks (NN) have been ubiquitously applied in the modeling of complex dynamical systems. Through a data collection and training procedure, NNs can capture accurate representations of the system dynamics even in challenging scenarios where high-speed aerodynamic effects [1], [2] or contact-rich environments [3], [4] are present. Moreover, NNs can be easily updated online as more data is collected, making them suitable for online tasks or modeling changing environments. For example, NN dynamical systems are widely used in model-based reinforcement learning [5] and learning-based adaptive control [2].

However, applying NN dynamics brings about significant challenges in providing safety guarantees for the controlled system. Benefiting from the expressivity of NNs, we are meanwhile faced with the high nonlinearity and large scale of NNs since they are often overparameterized. The runtime assurance (RTA) [6] mechanism provides a practical and effective solution to guarantee the safety of complex dynamical systems by designing a safety filter that primarily focuses on enforcing safety constraints. Given a primary controller that aims to optimize performance, the safety filter monitors and modifies the output of the primary controller online to guarantee that only safe control inputs are applied. The safety filter allows the design of the safety and performance-based controllers to be decoupled and has found wide applications in safe learning-based control [7], [8].

Shaoru Chen is with Microsoft Research, 300 Lafayette Street, New York, NY, 10012, USA, shaoruchen@microsoft.com. Kong Yao Chee, Nikolai Matni, M. Ani Hsieh, George J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, 19104, USA. {ckongyao, nmatni, m.hsieh, pappasg}@seas.upenn.edu.
Our codes are publicly available at https://github.com/ShaoruChen/NN-System-PSF.
† The first two authors contributed equally and are co-first authors.

In this work, we focus on the design of a predictive safety filter (PSF) [9] for uncertain NN dynamics. The PSF essentially follows a model predictive control (MPC) formulation with the nonlinear dynamics and constraints encoded in an optimization problem. Different from MPC, the PSF is less complex to solve since it does not consider any performance objectives [9]. Compared with the alternative safety filter construction schemes through control barrier functions (CBF) [10], [11] or Hamilton-Jacobi (HJ) reachability analysis [12], the PSF enjoys flexibility in handling dynamically changing NN models or model uncertainty bounds when updated online. We refer the interested readers to [9, Section 1 and 2] for a detailed discussion of the PSF, CBF, and HJ reachability-based safety filters.

Contributions: In this work, we consider uncertain NN dynamics subject to bounded additive disturbances, where the disturbances can encapsulate the errors between the learned NN dynamics and the true system. Despite being highly expressive, the considered uncertain NN dynamics requires solving a robust optimization problem involving NN dynamical constraints online in the PSF. To resolve this computational challenge, we propose to apply NN verification tools [13] to abstract the NN dynamics locally as a linear uncertain system, thereby reducing the original PSF problem into one that is amenable to robust linear MPC and convex optimization. In particular, we adapt the SLS (System Level Synthesis) MPC method [14] to solve the resulting robust MPC problem. A schematic of our pipeline is shown in Fig. 1. Soft constraints are used in robust linear MPC where slack variables denoting constraint violations are penalized. By applying a hierarchy of conservative function and model uncertainty approximations, we transform the original optimization problem into a convex one. A safety certificate for the uncertain NN dynamics over a finite horizon can then be provided when all slack variables are zero. Our contributions are summarized below.

1) Drawing tools from NN verification and robust linear MPC, we propose a novel predictive safety filter for uncertain NN dynamics through convex optimization. Importantly, the complexity of the convex optimization problem is independent of the NN size (i.e., width and depth of the NN).
2) Our PSF provides a safety certificate for the uncertain NN dynamics over a finite horizon when a certain numerical criterion is met by the convex optimization solutions.

A. Related works

The problem of ensuring the safety of learning-based systems has received significant interest with a plethora of methods described in [15]. Directly related to our work is the PSF developed in [9], which monitors and modifies a given control input by solving a predictive control problem online to guarantee the safety of the system. This formulation has been extended to the SLS setting [16], applied to racing cars [7] and a soft-constrained variant is proposed in [17] to handle unexpected disturbances to the states. The PSF that we propose differs from those in the existing work in the following ways. First, we exploit the structure of the neural networks to extract linear bounds on the NN outputs using NN verification tools [13], [18], simplifying the PSF formulation for NN dynamics. Second, our proposed pipeline circumvents the need to solve a robust non-convex optimization problem, even with the consideration of additive disturbances within the uncertain dynamics, as typical for nonlinear variants of SLS [19]. Unlike existing work in predictive control of NN dynamics [20]–[22], our work considers robust control of uncertain NN dynamics with a focus on obtaining formal safety guarantees.

Notation: $[N]$ denotes the set $\{0, 1, \dots, N\}$. $\mathcal{B}_p(z, r)$ denotes the ℓ_p ball centered at z with radius r. We use $x_{0:T}$ to denote the sequence $\{x_0, \dots, x_T\}$. $\mathbf{0}$ denotes a vector of all zeros, and its dimension can be inferred from context. For a vector $d \in \mathbb{R}^n$, $S = \text{diag}(d) \in \mathbb{R}^{n\times n}$ denotes a diagonal matrix with d being the diagonal vector. For a sequence of matrices $S_1, \cdots, S_N$, $S = \text{blkdiag}(S_1, \cdots, S_N)$ denotes a block diagonal matrix whose diagonal blocks are $S_1, \cdots, S_N$ arranged in the order. We represent a linear, causal operator $\mathbf{R}$ defined over a horizon T by the block-lower-triangular matrix

$$\mathbf{R} = \begin{bmatrix} R^{0,0} & & & \\ R^{1,1} & R^{1,0} & & \\ \vdots & \ddots & \ddots & \\ R^{T,T} & \cdots & R^{T,1} & R^{T,0} \end{bmatrix} \tag{1}$$

where $R^{i,j} \in \mathbb{R}^{p\times q}$ is a matrix of compatible dimension. The set of such matrices is denoted by $\mathcal{L}_{TV}^{T,p\times q}$ and the superscript T or $p \times q$ will be dropped when it is clear from the context.

II. Problem Formulation

Consider the following discrete-time nonlinear system,

$$x(k+1) = Ax(k) + Bu(k) + f\left(x(k), u(k)\right) + w(k), \tag{2}$$

where the vectors $x(k) \in \mathbb{R}^{n_x}$ and $u(k) \in \mathbb{R}^{n_u}$ are the state and control input. The vector $w(k) \in \mathbb{R}^{n_x}$ denotes the additive disturbances that can account for unknown effects from the environment or unmodeled dynamics. We assume that $w(k)$ is norm-bounded, *i.e.*, $w(k) \in \mathcal{W} := \{w \mid \|w\|_\infty \leq \sigma_w\}$. The system consists of a set of linear dynamics characterized by the matrices A and B and a set of nonlinear dynamics, $f(x(k), u(k))$. Specifically, these nonlinear dynamics are given by a NN $f : \mathbb{R}^{n_x+n_u} \mapsto \mathbb{R}^{n_x}$ with an arbitrary architecture. While we discuss our proposed approach taking into account a general formulation of dynamics in (2), the approach allows the matrices A and B to be zero and has the flexibility to account for time-varying dynamics $(A(k), B(k))$. The system (2) is required to satisfy the following constraints,

$$x(k) \in \mathcal{X}, \quad u(k) \in \mathcal{U}, \tag{3}$$

where $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$ are polytopes. The state $x(k)$ and input $u(k)$ are considered *safe* if they satisfy constraints (3).

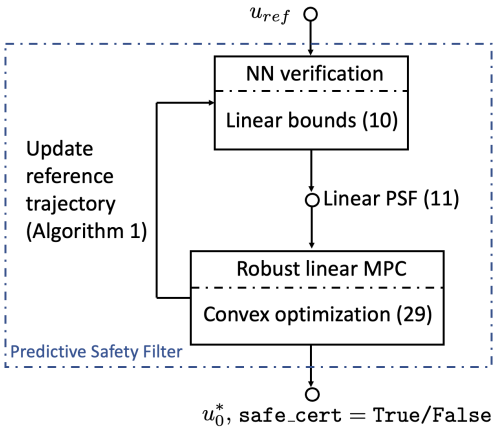

Fig. 1. Pipeline of the proposed predictive safety filter for uncertain NN systems.

A. Predictive safety filter

We assume a primary controller $\pi(x)$ which aims to complete a task or achieve high performance is given for system (2). Following the runtime assurance scheme, the primary controller $\pi(x)$ is not guaranteed to be safe since its design may be decoupled from the safety requirements. To ensure constraint satisfaction of the closed-loop system, we aim to design a predictive safety filter that monitors and modifies the control input $\pi(x(k))$ given by the primary controller in a minimally invasive manner online. This is achieved by solving the following robust optimization problem at each time step k,

$$\begin{aligned} \underset{u_{0:T-1}}{\text{minimize}} \quad & \|u_0 - \pi(x(k))\|_2^2 \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t + f(x_t, u_t) + w_t, \\ & x_t \in \mathcal{X}, u_t \in \mathcal{U}, \forall w_t \in \mathcal{W}, \quad t \in [T], \\ & x_0 = x(k), \end{aligned} \tag{4}$$

where the vectors x_t, u_t denote the predicted states and inputs over the horizon T, with $x(k)$ representing the current state of the system. We denote Problem (4) as the PSF problem and the sequence $u_{0:T-1}^*$ as the optimal solution of Problem (4). When applied, the control inputs $u_{0:T-1}^*$ can guarantee the safety of the system for the next T steps. In practice, the PSF problem (4) is solved recursively at each time step k, and the first optimal control input u_0^* is applied to the system, analogous to an MPC scheme.

B. Challenges with the predictive safety filter

While the solution to Problem (4) is able to provide safety guarantees, solving Problem (4) is a challenging task. Some potential issues include

(i) it is well known in robust MPC that searching over open-loop control sequences $u_{0:T-1}$ can be overly conservative [23],
(ii) the presence of the NN dynamics $f(x,u)$ makes solving the PSF computationally challenging,
(iii) the safety certificate of the solution $u^*_{0:T-1}$ is not available until convergence is reached, and
(iv) without the availability of a robust forward invariant set, attempting to solve the PSF may result in infeasibility.

To handle all the aforementioned issues, we combine NN verification tools [13] and robust MPC [14]. Our solution consists of two steps. First, we generate local linear bounds for the NN dynamics $f(x,u)$ using tools from NN verification. Next, we apply robust linear MPC to synthesize a state feedback control policy that guarantees robust constraint satisfaction for the system. This combined procedure provides a powerful simplification of the PSF problem and resolves issues (i) to (iii). To address the issue (iv), we introduce soft constraints in our formulation. This provides formal safety guarantees for the system when the slack variables are zero. We describe these two steps in Sections III and IV, respectively. Section V demonstrates our method numerically, and Section VI concludes the paper.

Remark 1: To ensure the safety constraints are satisfied *at all times* or guarantee recursive feasibility of the PSF (4), a local forward invariant set for the nonlinear system (2) is required, which is generally challenging to find. In this work, we do not assume the availability of such a forward invariant set and use slack variables as numerical certificates of safety. We leave the synthesis of the forward invariant terminal set for NN dynamics and its integration into the PSF as part of our future work.

III. Neural network verification bounds

In this section, we demonstrate how tools from NN verification can be utilized to over-approximate the NN dynamics with a linear time-varying (LTV) representation. This enables us to conservatively transform the PSF problem into a robust convex optimization problem, which is simpler to solve.

Given a bounded input set, the linear relaxation-based perturbation analysis (LiRPA) [13] is an efficient method to synthesize linear lower and upper bounds for the outputs of a NN with a general architecture. The bounds computed from this method are described in the following theorem.

Theorem 1: (rephrasing [18, Theorem 3.2]) Given a NN $f(z):\mathbb{R}^{n_0}\mapsto\mathbb{R}^{n_L}$, there exist two explicit linear functions $f_U:\mathbb{R}^{n_0}\mapsto\mathbb{R}^{n_L}$ and $f_L:\mathbb{R}^{n_0}\mapsto\mathbb{R}^{n_L}$ such that for all $z\in\mathcal{B}_p(z_0,r)$, we have

$$f_L(z)=A_Lz+b_L\le f(z)\le A_Uz+b_U=f_U(z), \quad (5)$$

where the inequalities are applied component-wise.

The parameters A_L, A_U, b_L, b_U are derived from the weights, biases and activation functions of the NN. In this paper, we choose $p:=\infty$ such that $\mathcal{B}_\infty(z,r)$ is polyhedral. The bounds (5) are computed using closed-form updates with a computational complexity polynomial in the number of neurons [18]. This allows the method to scale well to deep networks. However, if the NN is deep or if the input domain $\mathcal{B}_\infty(z,r)$ is large, the computed bounds tend to be loose. Motivated by this observation, we propose to extract a set of local linear bounds along a *reference trajectory*. Specifically, at every time step k, we construct a reference trajectory given by the sequences of reference states $\hat{\mathbf{x}}:=\hat{x}_{0:T}$ and control inputs $\hat{\mathbf{u}}:=\hat{u}_{0:T-1}$ where

$$\begin{aligned}&\hat{x}_{t+1}=A\hat{x}_t+B\hat{u}_t+f(\hat{x}_t,\hat{u}_t),\quad t\in[T-1],\\&\hat{x}_0=x(k).\end{aligned} \quad (6)$$

The reference control inputs $\hat{\mathbf{u}}$ can be obtained, e.g., by rolling out the nominal NN dynamics following the primary policy $\pi(\cdot)$. By denoting $z_t:=[x_t^\top\ u_t^\top]^\top$, $\hat{z}_t:=[\hat{x}_t^\top\ \hat{u}_t^\top]^\top$ and r_t to be the radius of the ℓ_∞ ball around $\hat{z}_t$, we apply Theorem 1 to get the following bounds for the NN dynamics along the reference trajectory,

$$\begin{bmatrix}\underline{G}_t^x & \underline{G}_t^u\end{bmatrix}z_t+\underline{g}_t\le f(x_t,u_t)\le\begin{bmatrix}\bar{G}_t^x & \bar{G}_t^u\end{bmatrix}z_t+\bar{g}_t, \quad (7)$$

for all $(x_t,u_t)\in\mathcal{B}_\infty(\hat{z}_t,r_t)$. In other words, the NN dynamics $f(x_t,u_t)$ is over-approximated with a set of linear lower and upper bounds. The ball $\mathcal{B}_\infty(\hat{z}_t,r_t)$ is referred to as the *trust region* in which the bounds (7) are valid.

To reduce conservatism in the formulation of the filter within the robust MPC framework, we integrate the bounds into the linear dynamics of the system. Specifically, using the bounds in (7), we define

$$f_d(x_t,u_t):=f(x_t,u_t)-\left(\tilde{A}_tx_t+\tilde{B}_tu_t+\tilde{c}_t\right), \quad (8)$$

where

$$\tilde{A}_t:=\frac{\underline{G}_t^x+\bar{G}_t^x}{2},\ \tilde{B}_t:=\frac{\underline{G}_t^u+\bar{G}_t^u}{2},\ \tilde{c}_t:=\frac{\underline{g}_t+\bar{g}_t}{2} \quad (9)$$

denote the means of the linear bounds.

Corollary 1: Given the bounds in (7), for every $z_t:=[x_t^\top\ u_t^\top]^\top\in\mathcal{B}_\infty(\hat{z}_t,r_t)$, the dynamics $f_d(x_t,u_t)$ in (8) have the following bounds,

$$\begin{bmatrix}\underline{D}_t^x & \underline{D}_t^u\end{bmatrix}z_t+\underline{d}_t\le f_d(x_t,u_t)\le\begin{bmatrix}\bar{D}_t^x & \bar{D}_t^u\end{bmatrix}z_t+\bar{d}_t, \quad (10)$$

where

$$\underline{D}_t^x=\frac{\underline{G}_t^x-\bar{G}_t^x}{2}=-\bar{D}_t^x,\ \underline{D}_t^u=\frac{\underline{G}_t^u-\bar{G}_t^u}{2}=-\bar{D}_t^u,$$
$$\underline{d}_t=\frac{\underline{g}_t-\bar{g}_t}{2}=-\bar{d}_t.$$

It is important to note that although the NN dynamics $f(x_t,u_t)$ can have large values within the trust region $\mathcal{B}_\infty(\hat{z}_t,r_t)$, the dynamics $f_d(x_t,u_t)$ tends to be small in magnitude and can be treated as disturbances to the system. With the extraction of $f_d(x_t,u_t)$, we obtain an LTV reformulation of the PSF problem, referred to as the *linear*

PSF problem,

$$\begin{aligned}\underset{u_{0:T-1}}{\text{minimize}} \quad & \|u_0 - \pi(x(k))\|_2^2 \\ \text{subject to} \quad & x_{t+1} = A_t x_t + B_t u_t + c_t + \Delta_t(x_t, u_t) + w_t, \\ & (x_t, u_t) \in \mathcal{B}_\infty(\hat{z}_t, r_t), \quad t \in [T-1], \\ & x_t \in \mathcal{X},\ u_t \in \mathcal{U}, \quad t \in [T], \\ & \forall \Delta_t(\cdot) \in \mathcal{P}_t,\ w_t \in \mathcal{W}, \quad t \in [T], \\ & x_0 = x(k),\end{aligned} \tag{11}$$

where the means in (9) are merged into the linear dynamics of the system with the definition of the following time-varying system parameters [1],

$$A_t := A + \tilde{A}_t,\ B_t := B + \tilde{B}_t,\ c_t := \tilde{c}_t.$$

The uncertainty set $\mathcal{P}_t$ is given as

$$\mathcal{P}_t := \left\{ \Delta_t(x_t, u_t) \middle| \begin{array}{l} \Delta_t(x_t, u_t) \geq \begin{bmatrix} \underline{D}_t^x & \underline{D}_t^u \end{bmatrix} z_t + \underline{d}_t \\ \Delta_t(x_t, u_t) \leq \begin{bmatrix} \bar{D}_t^x & \bar{D}_t^u \end{bmatrix} z_t + \bar{d}_t \end{array} \right\}, \tag{12}$$

using the bounds obtained from Corollary 1. It immediately follows that $f_d(x_t, u_t)$ corresponds to a realization of $\Delta_t(x_t, u_t)$.

A few remarks about the linear PSF problem are in order. First, the solution of the linear PSF problem is dependent on the centers $\hat{\mathbf{z}} = \hat{z}_{0:T}$ and radius r_t of the trust regions. The centers play an important role when the reference trajectory lies near or beyond the boundaries of the constraint set. In this case, the reference trajectory should be shifted towards the constraint set and this is done by adjusting the centers of these trust regions. Next, based on how the linear bounds in (12) are computed, there is a trade-off in the size of the radius r_t. A small radius ensures that the computed bounds are accurate, but it limits the range in which the centers $\hat{\mathbf{z}}$ can be updated at each iteration. On the other hand, a large radius provides more flexibility in updating the reference trajectory, but the bounds can be overly conservative. Lastly, any feasible solution $u^*_{0:T-1}$ to the linear PSF problem (11) is also feasible for the PSF problem (4).

With these considerations, we devise a method such that the trust regions $\mathcal{B}_\infty(\hat{z}_t,\ r_t)$ can be updated online. Details of this update are given in Section IV-E. In Section IV-A to IV-D, we first describe how to solve the linear PSF problem through robust linear MPC.

IV. Robust Linear MPC

Compared with the PSF problem (4), the linear PSF problem (11) only involves uncertain linear dynamics. However, solving it can still be a challenge and a conservative approach may not succeed. Since optimizing over open-loop control sequences is conservative in robust MPC, we consider optimizing over state-feedback controllers $u_t = \mu_t(x_{0:t})$ instead. To achieve this, we apply an extension of the SLS MPC algorithm in [14] which is shown to enjoy outstanding tightness among the existing robust linear MPC methods.

[1]When a time-varying dynamics $(A(k), B(k))$ is considered in (2), we can replace (A, B) by their time-varying counterparts in the definitions of (A_t, B_t).

A. Overview of SLS MPC

In SLS MPC, we consider the following uncertain linear time-varying system,

$$x_{t+1} = A_t x_t + B_t u_t + \xi_t \tag{13}$$

where the time-varying matrices that represent the nominal dynamics (A_t, B_t) are known and ξ_t denotes the lumped uncertainty, which will be used to capture the effects of uncertainty in the dynamics.

Consider the dynamics (13) over a horizon T. We define the following variables, which are concatenations of the variables in (13) over the horizon T,

$$\begin{aligned} &\mathbf{x} := [x_0^\top \ \cdots \ x_T^\top]^\top, \qquad \mathbf{u} := [u_0^\top \ \cdots \ u_T^\top]^\top, \\ &\boldsymbol{\xi} := [x_0^\top \ \xi_0^\top \ \cdots \ \xi_{T-1}^\top]^\top, \end{aligned} \tag{14}$$

and these concatenated system matrices,

$$\mathbf{A} := \text{blkdiag}\left(A_0, \cdots, A_T\right), \mathbf{B} := \text{blkdiag}\left(B_0, \cdots, B_T\right).$$

We define Z as the block-downshift operator with the first block sub-diagonal filled with identity matrices and zeros everywhere else. The dynamics (13) over the horizon T can then be compactly written as,

$$\mathbf{x} = Z\mathbf{A}\mathbf{x} + Z\mathbf{B}\mathbf{u} + \boldsymbol{\xi}, \tag{15}$$

Next, we consider a LTV state feedback controller $u_t = \sum_{i=0}^{t} K^{t,t-i} x_i$, compactly given as $\mathbf{u} = \mathbf{K}\mathbf{x}$, $\mathbf{K} \in \mathcal{L}_{TV}^{T, n_u \times n_x}$. Plugging $\mathbf{u}$ into (15) gives the following *system responses* $\{\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u\}$, mapping $\boldsymbol{\xi}$ to the closed-loop states and inputs $(\mathbf{x}, \mathbf{u})$,

$$\begin{bmatrix} \boldsymbol{\Phi}_x \\ \boldsymbol{\Phi}_u \end{bmatrix} := \begin{bmatrix} (I - Z(\mathbf{A} + \mathbf{B}\mathbf{K}))^{-1} \\ \mathbf{K}(I - Z(\mathbf{A} + \mathbf{B}\mathbf{K}))^{-1} \end{bmatrix}. \tag{16}$$

The following theorem establishes the connection between $\{\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u\}$ and state feedback controllers.

Theorem 2: [24, Theorem 2.1] Over the horizon $t \in [T]$, for the dynamics (13) with the LTV state feedback controller $\mathbf{u} = \mathbf{K}\mathbf{x}$, $\mathbf{K} \in \mathcal{L}_{TV}^{T, n_u \times n_x}$, we have:

1) The affine subspace defined by

$$\begin{bmatrix} I - Z\mathbf{A} & -Z\mathbf{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_x \\ \boldsymbol{\Phi}_u \end{bmatrix} = I,\ \boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u \in \mathcal{L}_{TV}^T \tag{17}$$

parameterizes all possible system responses (16).

2) For any $\{\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u\} \in \mathcal{L}_{TV}^T$ satisfying (17), the controller gain $\mathbf{K} = \boldsymbol{\Phi}_u \boldsymbol{\Phi}_x^{-1} \in \mathcal{L}_{TV}^T$ achieves the desired responses (16).

The system responses explicitly characterize the effects of the lumped uncertainty $\boldsymbol{\xi}$ on $(\mathbf{x}, \mathbf{u})$. In a previous work [14], a system subjected to polytopic uncertainties and additive disturbances is considered, *i.e.*, $\xi_t := \Delta_A x_t + \Delta_B u_t + w_t$ where the parameters (Δ_A, Δ_B) belong to a polytopic set. Interested readers are referred to [14] for more details on SLS MPC.

Despite its outstanding performance in conservatism reduction compared to existing robust MPC methods [14], applying SLS MPC directly to solve the linear PSF comes with these two challenges: (i) the uncertainty set $\mathcal{P}_t$ is defined through affine inequalities, and converting $\mathcal{P}_t$ into a vertex representation, which is amenable to existing robust

MPC methods [25]–[28], requires 2^{n_x} vertices; (ii) applying SLS MPC requires merging the constant c_t into the lumped uncertainty ξ_t which causes the bounds on $\boldsymbol{\xi}$ to be overly conservative. In the following subsections, we describe an extension to SLS MPC to address these two challenges.

B. Controller parameterization

For the uncertain linear dynamics

$$x_{t+1} = A_t x_t + B_t u_t + c_t + \Delta_t(x_t, u_t) + w_t \tag{18}$$

stated in (11), we define $\eta_t := \Delta_t(x_t, u_t) + w_t$ as the lumped uncertainty. Instead of treating $\xi_t = c_t + \eta_t$ in (13), we decompose (18) into a set of nominal and error dynamics.

First, in addition to (14), we concatenate these variables over the horizon T,

$$\begin{aligned}&\boldsymbol{\eta} := [\mathbf{0}^\top \ \eta_0^\top \ \cdots \ \eta_{T-1}^\top]^\top, \ \mathbf{w} := [\mathbf{0}^\top \ w_0^\top \ \cdots \ w_{T-1}^\top]^\top,\\ &\mathbf{c} := [\mathbf{0}^\top c_0^\top \ \cdots \ c_{T-1}^\top]^\top, \quad \boldsymbol{\delta}_{x_0} := [x_0^\top \ \mathbf{0}^\top \ \cdots \ \mathbf{0}^\top]^\top.\end{aligned} \tag{19}$$

We define the nominal and error states $\{\mathbf{h}, \mathbf{x}_e\}$ and control inputs $\{\mathbf{v}, \mathbf{u}_e\}$ as

$$\begin{aligned}\mathbf{h} &:= [h_0^\top \ \cdots \ h_T^\top]^\top, \quad \mathbf{v} := [v_0^\top \ \cdots \ v_T^\top]^\top,\\ \mathbf{x}_e &:= [x_{e,0}^\top \ \cdots \ x_{e,T}^\top]^\top = \mathbf{x} - \mathbf{h},\\ \mathbf{u}_e &:= [u_{e,0}^\top \ \cdots \ u_{e,T}^\top]^\top = \mathbf{u} - \mathbf{v},\end{aligned}$$

with the nominal and error dynamics as

$$\mathbf{h} = Z(\mathbf{A}\mathbf{h} + \mathbf{B}\mathbf{v}) + \mathbf{c} + \boldsymbol{\delta}_{x_0}, \tag{20a}$$

$$\mathbf{x}_e = Z(\mathbf{A}\mathbf{x}_e + \mathbf{B}\mathbf{u}_e) + \boldsymbol{\eta}. \tag{20b}$$

It is important to note that (20b) conforms with (13). A LTV state feedback controller $\mathbf{K} \in \mathcal{L}_{TV}^{T, n_u \times n_x}$ is then applied to control the error states. The overall controller for (18) is given by $\mathbf{u} = \mathbf{K}\mathbf{x}_e + \mathbf{v} = \mathbf{K}(\mathbf{x} - \mathbf{h}) + \mathbf{v}$.

C. Lumped uncertainty over-approximation

For the lumped uncertainty $\boldsymbol{\eta}$, its dependence on $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{w}$ complicates the design of the robust controller. As in SLS MPC, the approach is to over-approximate $\boldsymbol{\eta}$ by an independent, filtered virtual disturbance signal $\boldsymbol{\Psi}\widetilde{\mathbf{w}}$, where $\widetilde{\mathbf{w}} = [\mathbf{0}^\top \ \tilde{w}_0^\top \ \cdots \ \tilde{w}_{T-1}^\top]^\top$, $\|\tilde{w}_t\|_\infty \leq 1$. The matrix $\boldsymbol{\Psi} \in \mathcal{L}_{TV}^{T, n_x \times n_x}$ is a filter operating on the finite-horizon virtual disturbance signal $\widetilde{\mathbf{w}}$, with its diagonal blocks of $\boldsymbol{\Psi}$ structured as $\Psi^{0,0} = I, \Psi^{t,0} = \text{diag}(\psi_{t-1})$ with $\psi_{t-1} \in \mathbb{R}^{n_x}, \psi_{t-1} > 0, t = 1, \ldots, T$.

We define $\widetilde{\mathcal{W}} = \{\widetilde{\mathbf{w}} \mid \|\tilde{w}_t\|_\infty \leq 1, t \in [T-1]\}$ as the set of admissible virtual disturbances. Since $\tilde{w}_t$ are unit norm-bounded, we tune the filter $\boldsymbol{\Psi}$ to change the reachable set of $\boldsymbol{\Psi}\widetilde{\mathbf{w}}$, defined as $\mathcal{R}(\boldsymbol{\Psi}\widetilde{\mathbf{w}}) := \{\boldsymbol{\zeta} \mid \boldsymbol{\zeta} = \boldsymbol{\Psi}\widetilde{\mathbf{w}}, \widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}\}$, $\boldsymbol{\zeta} := [\mathbf{0}^\top \ \zeta_0^\top \ \cdots \ \zeta_{T-1}^\top]^\top$.

Our goal is to find sufficient conditions on the control parameters $\{\mathbf{K}, \mathbf{h}, \mathbf{v}\}$ and $\boldsymbol{\Psi}$ such that the reachable set of $\boldsymbol{\eta}$, denoted by $\mathcal{R}(\boldsymbol{\eta}; \mathbf{K}, \mathbf{h}, \mathbf{v}) := \{\boldsymbol{\eta} \mid \eta_t = \Delta_t(x_t, u_t) + w_t, \Delta_t \in \mathcal{P}_t, t \in [T-1]\}$, is a subset of the reachable set of $\boldsymbol{\Psi}\widetilde{\mathbf{w}}$. The following proposition provides these sufficient conditions and the proof is given in the appendix of the extended version [29] of this paper.

Proposition 1: Let $e_i \in \mathbb{R}^{n_x}$ denote the i-th standard basis for $i = 1, \ldots, n_x$ and $y_t = [h_t^\top \ v_t^\top]^\top$, $\Phi^{t,t-i} := \begin{bmatrix}\Phi_x^{t,t-i\top} & \Phi_u^{t,t-i\top}\end{bmatrix}^\top$ for $t \in [T-1]$. The following constraints:

$$\begin{bmatrix} I - Z\mathbf{A} & -Z\mathbf{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}_x \\ \boldsymbol{\Phi}_u\end{bmatrix} = \boldsymbol{\Psi}, \ \boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u \in \mathcal{L}_{TV}^T, \tag{21}$$

and

$$\sigma_w + e_i^\top([\bar{D}_0^x \ \bar{D}_0^u]y_0 + \bar{d}_0) \leq \psi_{0,i}, \tag{22a}$$

$$\sigma_w - e_i^\top([\underline{D}_0^x \ \underline{D}_0^u]y_0 + \underline{d}_0) \leq \psi_{0,i}, \tag{22b}$$

$$\sigma_w + e_i^\top\Big([\bar{D}_t^x \ \bar{D}_t^u]y_t + \bar{d}_t\Big) + \sum_{i=1}^t \left\| e_i^\top\Big([\bar{D}_t^x \ \bar{D}_t^u]\Phi^{t,t-i} - \Psi^{t+1,t+1-i}\Big)\right\|_1 \leq \psi_{t,i}, \tag{22c}$$

$$\sigma_w - e_i^\top\Big([\underline{D}_t^x \ \underline{D}_t^u]y_t + \underline{d}_t\Big) + \sum_{i=1}^t \left\| e_i^\top\Big([\underline{D}_t^x \ \underline{D}_t^u]\Phi^{t,t-i} - \Psi^{t+1,t+1-i}\Big)\right\|_1 \leq \psi_{t,i}, \tag{22d}$$

$$i \in [n_x], \quad t = 1, \cdots, T-1,$$

guarantee that $\mathcal{R}(\boldsymbol{\eta}; \mathbf{K}, \mathbf{h}, \mathbf{v}) \subseteq \mathcal{R}(\boldsymbol{\Psi}\widetilde{\mathbf{w}})$ holds.

D. Convex formulation of the linear PSF

With the constraints (21) and (22), we have for any realization of $\boldsymbol{\eta}$, there exists $\widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}$ such that $\boldsymbol{\eta} = \boldsymbol{\Psi}\widetilde{\mathbf{w}}$. Therefore, we can represent $\boldsymbol{\eta}$ as $\boldsymbol{\Psi}\widetilde{\mathbf{w}}$ and write the error dynamics (20b) as

$$\mathbf{x}_e = Z(\mathbf{A}\mathbf{x}_e + \mathbf{B}\mathbf{u}_e) + \boldsymbol{\Psi}\widetilde{\mathbf{w}}. \tag{23}$$

By [14, Corollary 1], we have constraint (21) parameterizes all system responses $\mathbf{x}_e = \boldsymbol{\Phi}_x\widetilde{\mathbf{w}}, \mathbf{u}_e = \boldsymbol{\Phi}_u\widetilde{\mathbf{w}}$ of system (23) under the controller $\mathbf{u}_e = \mathbf{K}\mathbf{x}_e$. Then, the closed-loop states and control inputs of system (18) are given by

$$\mathbf{x} = \mathbf{h} + \boldsymbol{\Phi}_x\widetilde{\mathbf{w}}, \quad \mathbf{u} = \mathbf{v} + \boldsymbol{\Phi}_u\widetilde{\mathbf{w}}. \tag{24}$$

To guarantee robust constraint satisfaction of the controller $\mathbf{u} = \mathbf{K}(\mathbf{x} - \mathbf{h}) + \mathbf{v}$, we tighten the constraints in the linear PSF (11). Consider the state constraint $x_t \in \mathcal{X}$ as an example. The constraints are represented by a polyhedral set, $\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid F_x x \leq b_x\}$, where $F_x \in \mathbb{R}^{n_\mathcal{X} \times n_x}$, $b_x := [b_1^x, \ldots, b_{n_\mathcal{X}}^x]^\top \in \mathbb{R}^{n_\mathcal{X}}$, and $\{a_j^{x\top}\}_{j=1}^{n_\mathcal{X}}$ denote the rows of F_x. This implies that (a_j^x, b_j^x) denotes the j-th set of linear constraint parameters in $\mathcal{X}$. From (24), we have $x_t = h_t + \sum_{i=1}^t \Phi_x^{t,t-i}\tilde{w}_{i-1}$. Then, the following constraints

$$a_j^{x\top}h_t + \sum_{i=1}^t \left\| a_j^{x\top}\Phi_x^{t,t-i}\right\|_1 \leq b_j^x, \quad j = 1, \ldots, n_\mathcal{X} \tag{25}$$

guarantee that all constraints in $\mathcal{X}$ are satisfied robustly. As discussed in Section II, the recursive feasibility of the linear PSF cannot be guaranteed without a robust forward invariant set. Therefore, in this work, we introduce soft constraints into (25),

$$a_j^{x\top}h_t + \sum_{i=1}^t \left\| a_j^{x\top}\Phi_x^{t,t-i}\right\|_1 \leq b_j^x + \epsilon_{x,j}^t, \quad \epsilon_{x,j}^t \geq 0, \ t \in [T], \ j = 1, \ldots, n_\mathcal{X}. \tag{26}$$

In the cost function, a large penalty on $\|\epsilon_x^t\|_1$ is applied. In the case $\epsilon_{x,j}^t = 0$, we obtain robust constraint satisfaction

guarantee for the constraint with parameters (a_j^x, b_j^x). Similarly, the input constraints can be tightened as

$$a_k^{u\top} v_t + \sum_{i=1}^{t} \left\| a_k^{u\top} \Phi_u^{t,t-i} \right\|_1 \le b_k^u + \epsilon_{u,k}^t, \quad \epsilon_{u,k}^t \ge 0,\ t \in [T], \quad k = 1, \dots, n_{\mathcal{U}}, \tag{27}$$

where $n_{\mathcal{U}}$ denotes the number of linear inequalities defining $\mathcal{U}$ and (a_k^u, b_k^u) denotes the k-th set of linear constraint parameters in $\mathcal{U}$. The trust region constraints $(x_t, u_t) \in \mathcal{B}_\infty(\hat{z}_t, r_t)$ analogously can be tightened as

$$\begin{aligned} &a_j^\top h_t + \sum_{i=1}^{t} \left\| a_j^\top \Phi_x^{t,t-i} \right\|_1 \le b_j + \sigma_{x,j}^t,\ \sigma_{x,j}^t \ge 0, \\ &a_k^\top v_t + \sum_{i=1}^{t} \left\| a_k^\top \Phi_u^{t,t-i} \right\|_1 \le b_k + \sigma_{u,k}^t,\ \sigma_{u,k}^t \ge 0, \\ &j = 1, \dots, 2n_x,\ k = 1, \dots, 2n_u,\ t \in [T-1], \end{aligned} \tag{28}$$

where (a_j, b_j) and (a_k, b_k) are defined similarly for the state and input-related constraint parameters in $\mathcal{B}_\infty(\hat{z}_t, r_t)$.

Summarizing the results above, we propose a convex tightening of the linear PSF, which can be written as

$$\begin{aligned} \underset{\substack{\mathbf{\Phi}_x, \mathbf{\Phi}_u, \mathbf{\Psi}, \mathbf{h}, \mathbf{v}, \\ \{\epsilon_x^t, \epsilon_u^t, \sigma_x^t, \sigma_u^t\}}}{\text{minimize}} \quad & \|u_0 - \pi(x(k))\|_2^2 + M_\epsilon \sum_{t=0}^{T} \left(\left\|\epsilon_x^t\right\|_1 + \left\|\epsilon_u^t\right\|_1 \right) \\ & + M_\sigma \sum_{t=0}^{T-1} \left(\left\|\sigma_x^t\right\|_1 + \left\|\sigma_u^t\right\|_1 \right) \\ \text{subject to} \quad & \text{affine constraint (21)}, \\ & \text{over-approximation constraints (22)}, \\ & \text{soft state and input constraints (26) and (27)}, \\ & \text{soft trust region constraints (28)}, \\ & x_0 = x(k), \end{aligned} \tag{29}$$

where $M_\epsilon, M_\sigma > 0$ are chosen as large numbers. When a polyhedral robust forward invariant set is used as the terminal constraint, we can tighten it similarly as (26). All the constraints in Problem (29) are linear, making (29) a quadratic program. With the use of soft constraints, (29) is always feasible. If the slack variables $\{\epsilon_x^t, \epsilon_u^t, \sigma_x^t, \sigma_u^t\}$ in the solution are zero, we obtain a certificate that the system is safe for the next T steps under the state-feedback controller $\mathbf{u} = \mathbf{K}(\mathbf{x} - \mathbf{h}) + \mathbf{v}$ with $\mathbf{K} = \mathbf{\Phi}_u \mathbf{\Phi}_x^{-1}$.

E. Trust region update

Following the discussion on the trust regions in Section III, we describe a method to update the trust regions online. Starting with the reference trajectory given by the primary policy $\pi(x)$, we propose to iteratively increase r_t and update the reference trajectory by applying the policy $\mathbf{u} = \mathbf{K}(\mathbf{x} - \mathbf{h}) + \mathbf{v}$, synthesized in Section IV. We then pick the reference trajectory that gives the smallest slack variables and apply the corresponding control inputs to the system. Our framework is summarized in Algorithm 1.

Algorithm 1 Robust Linear MPC-based PSF

Input: Current state $x(k)$, horizon T, number of iterations N, initial reference trajectory $\hat{z}_{0:T}$, reference control input $\pi(x(k))$, initial trust region radius $r_0 > 0$

Output: Filtered control input u_0^*, safety certificate `safe_cert`

At each time step k,
Initialize: $r_t \leftarrow r_0,\ t \in [T-1], \{\mathbf{K}^*, \mathbf{h}^*, \mathbf{v}^*\} \leftarrow \{\mathbf{0}\}$, $\epsilon^* \leftarrow \infty$.
for $\ell = 1, \dots, N$ **do**
  Construct $\mathcal{B}_\infty(\hat{z}_t, r_t)$ and $\mathcal{P}_t$ with (7) and (10)
  Solve (29) with $x(k)$ to get $\{\mathbf{K}, \mathbf{h}, \mathbf{v}\}$
  Extract $\epsilon_{max} :=$ max. value of slack variables
  if $\epsilon_{max} = 0$ **then**
    Set $\{\mathbf{K}^*, \mathbf{h}^*, \mathbf{v}^*\} \leftarrow \{\mathbf{K}, \mathbf{h}, \mathbf{v}\}$
    Set `safe_cert` $\leftarrow$ `True`
  else
    Update $\hat{z}_{0:T}$ with $\mathbf{u} = \mathbf{K}(\mathbf{x} - \mathbf{h}) + \mathbf{v}$ ▷ see (6)
    Update $r_t \leftarrow \beta r_t$ for $t \in [T-1]$ with $\beta > 1$
    if $\epsilon_{max} \le \epsilon^*$ **then**
      Set $\epsilon^* \leftarrow \epsilon_{max}$
      Set $\{\mathbf{K}^*, \mathbf{h}^*, \mathbf{v}^*\} \leftarrow \{\mathbf{K}, \mathbf{h}, \mathbf{v}\}$
      Set `safe_cert` $\leftarrow$ `False`
Set $u_0^* \leftarrow v_0$, extracted from $\{\mathbf{K}^*, \mathbf{h}^*, \mathbf{v}^*\}$

V. Numerical Example

To verify the efficacy of the proposed solution, we test it on a NN proxy of the nonlinear pendulum system. The pendulum consists of the following dynamics [30],

$$\ddot{\theta} = \frac{3g \sin(\theta)}{2l} + \frac{3\tau}{ml^2}, \tag{30}$$

where θ is the angle between the pendulum and the vertical, m and l are the mass and length of the pendulum, g is the gravitational force, and τ is the external torque acting on the pendulum. The state and control input are defined as $x := [\theta\ \dot{\theta}]^\top \in \mathbb{R}^2$ and $u := \tau \in \mathbb{R}$. To obtain the linear dynamics in (2), the dynamics (30) are linearized about the origin and discretized with a sampling time of 0.05s to obtain the following dynamics,

$$x_{t+1} = \begin{bmatrix} 1.0092 & 0.05015 \\ 0.369 & 1.0092 \end{bmatrix} x_t + \begin{bmatrix} 0.00125 \\ 0.05015 \end{bmatrix} u_t. \tag{31}$$

Next, we train a NN $f(x, u)$ to approximate the residual dynamics that are not captured by the linear dynamics in (31). We first collect data by simulating the nonlinear dynamics in (30) for a duration of 15s. The NN is then trained through a backpropagation procedure, using the mean squared errors between the predicted and true states as the loss function. The NN consists of 3 hidden layers with 64 neurons in each layer and uses the rectified linear unit (ReLU) as the activation function. Additive noise with a maximum magnitude σ_w of $\{0.05, 0.1\}$ is injected into the states of the system. For the primary control policy, we consider an iterative linear quadratic regulator (iLQR) scheme [31] with the box-constrained heuristic [32]. This is

TABLE I

INITIAL CONDITIONS AND ANGLES FOR THE TEST SCENARIOS.

Test Case	x_0 [deg; deg/s]	$\theta_{r,1}$ [deg]	$\theta_{r,2}$ [deg]
1	[57.3; -120.3]	120	-50
2	[-85.9; -85.9]	-150	40
3	[-85.9; -114.6]	-100	-180
4	[85.9; 57.3]	100	180

implemented with the *mpc.pytorch* library [33]. The box-constrained heuristic allows the system to adhere to the control constraints, but does not account for state constraints.

Four test cases are considered in our experiments. In each of these cases, the system is required to track a pair of reference angles $(\theta_{r,1}, \theta_{r,2})$ sequentially, starting from an initial condition x_0 and across a duration of 2s. The initial conditions and reference angles are given in Table I. We simulate each of these test cases under 4 control schemes - (i) a nominal iLQR framework, (ii) a soft-constrained iLQR framework (SC-iLQR), (iii) safe-filtered iLQR, where we apply the proposed safety filter to the nominal iLQR scheme, and (iv) safe-filtered SC-iLQR, where the safety filter is applied to SC-iLQR. For SC-iLQR, soft state constraints are incorporated into the cost function of the forward pass of the iLQR algorithm. Specifically, the function $\phi(x) = \max\{0, x\}$ is applied onto each of the constraints, which increases the cost function proportionally whenever the constraints are violated. In safe-filtered iLQR and SC-iLQR, the initial reference trajectories $\hat{z}_{0:T}$ in Algorithm 1 were initialized by iLQR and SC-iLQR, respectively.

The state trajectories under these control schemes are plotted in Fig. 2 and the percentages of constraint violations are tabulated in Table II. These percentages are computed by taking the ratio between the number of points in which the states violate the constraints against the total number of points in the state trajectory. Since the nominal iLQR method does not account for state constraints, it results in the largest percentage of constraint violation. While the soft-constrained iLQR reduces the level of constraint violation, there are a number of instances where the constraints are violated, as depicted in Fig. 2. On the other hand, as shown in Table II, through the application of the safety filter to iLQR and SC-iLQR, no constraint violations are observed for the test cases. To illustrate the safety certificate obtained, we plot the slack variables that characterize the trust region and state and input constraints, together with the safety certificate for the third test case, under a maximum noise level $\sigma_w = 0.05$, in Fig. 3. The combination of Table II and Fig. 3 indicates that given the formulation in (29), the PSF may not give a numerical safety certificate even when the state trajectories are safe. Meanwhile, this proposed PSF can effectively encourage the system to behave safely by minimizing the conservative upper bounds on the constraint violation.

The statistics of the computational times of the control scheme, in this case the iLQR scheme, and the safety filter are depicted in Fig. 4. While introducing soft constraints increases the run times of the iLQR scheme, it has a significant effect on the run times of the safety filter, as

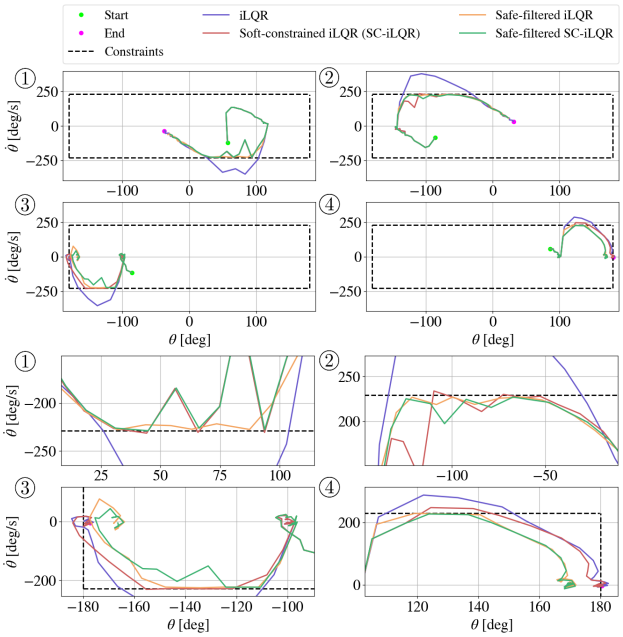

Fig. 2. **Top panel**: Plots of the state trajectories under the 4 test cases. The initial and final states are marked with green and magenta circles. The state constraints are plotted with black dashed lines. **Bottom panel**: Zoomed-in plots of the state trajectories, around the boundaries where constraint violation potentially occurs.

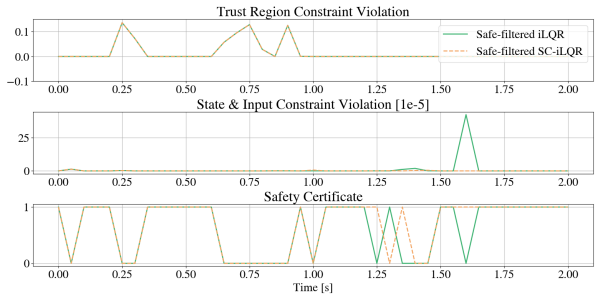

Fig. 3. **Top two panels**: Time histories of the slack variables for the trust region, state and input constraints. **Bottom panel**: Time histories of the safety certificate attained. The values of 1 and 0 denote `True` and `False` respectively.

observed in the right panel of Fig. 4. With the soft constraints added to the iLQR scheme, the state trajectories are closer to the boundaries of the constraint sets, without the activation of the safety filter. This allows the filter to find a solution that satisfies the constraints under a smaller number of iterations, which reduces computational time.

VI. CONCLUSION

We proposed a convex optimization-based predictive safety filter for uncertain NN dynamical systems with additive disturbances. By utilizing tools from NN verification and robust linear MPC, our method requires solving a soft-constrained convex program online whose complexity is independent of the NN size. With our framework, formal safety guarantees, together with a robust state-feedback controller, are attained when the slack variables in the solution are all zero.

TABLE II

PERCENTAGE OF STATE CONSTRAINT VIOLATIONS.

Method	Maximum noise level, $\sigma_w = 0.05$			
	Case 1 [%]	Case 2 [%]	Case 3 [%]	Case 4 [%]
iLQR	12.20	14.63	14.63	21.95
SC-iLQR	7.32	4.88	12.20	12.20
Safe-filtered iLQR	**0.0**	**0.0**	**0.0**	**0.0**
Safe-filtered SC-iLQR	**0.0**	**0.0**	**0.0**	**0.0**
	Maximum noise level, $\sigma_w = 0.1$			
	Case 1 [%]	Case 2 [%]	Case 3 [%]	Case 4 [%]
iLQR	14.63	14.63	17.07	26.83
SC-iLQR	4.87	2.44	7.32	9.75
Safe-filtered iLQR	**0.0**	**0.0**	**0.0**	**0.0**
Safe-filtered SC-iLQR	**0.0**	**0.0**	**0.0**	**0.0**

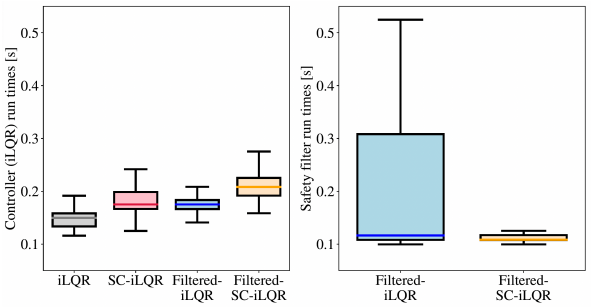

Fig. 4. Statistics of the run times of the controller and the safety filter under the four test methods.

REFERENCES

[1] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 4653–4660, IEEE, 2016.

[2] M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, p. eabm6597, 2022.

[3] Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, and V. Sindhwani, "Data efficient reinforcement learning for legged robots," in *Conference on Robot Learning*, pp. 1–10, PMLR, 2020.

[4] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic MPC for model-based reinforcement learning," in *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1714–1721, IEEE, 2017.

[5] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 7559–7566, IEEE, 2018.

[6] K. L. Hobbs, M. L. Mote, M. C. Abate, S. D. Coogan, and E. M. Feron, "Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems," *IEEE Control Systems Magazine*, vol. 43, no. 2, pp. 28–65, 2023.

[7] 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, pp. 7635–7642, 2021.

[8] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[9] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[10] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[11] H. Ma, B. Zhang, M. Tomizuka, and K. Sreenath, "Learning differentiable safety-critical control using control barrier functions for generalization to novel environments," in *2022 European Control Conference (ECC)*, pp. 1301–1308, IEEE, 2022.

[12] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[13] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, "Automatic perturbation analysis for scalable certified robustness and beyond," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1129–1141, 2020.

[14] S. Chen, V. M. Preciado, M. Morari, and N. Matni, "Robust model predictive control with polytopic model uncertainty through system level synthesis," *arXiv preprint arXiv:2203.11375*, 2022.

[15] 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, pp. 411–444, 2022.

[16] A. P. Leeman, J. Köhler, S. Bennani, and M. N. Zeilinger, "Predictive safety filter using system level synthesis," in *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, pp. 1180–1192, PMLR, 2023.

[17] K. P. Wabersich and M. N. Zeilinger, "Predictive control barrier functions: Enhanced safety mechanisms for learning-based control," *IEEE Transactions on Automatic Control*, 2022.

[18] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," *Advances in neural information processing systems*, vol. 31, 2018.

[19] A. P. Leeman, J. Köhler, A. Zanelli, S. Bennani, and M. N. Zeilinger, "Robust nonlinear optimal control via system level synthesis," *arXiv preprint arXiv:2301.04943*, 2023.

[20] A. Saviolo, G. Li, and G. Loianno, "Physics-inspired temporal learning of quadrotor dynamics for accurate model predictive trajectory tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10256–10263, 2022.

[21] N. A. Spielberg, M. Brown, and J. C. Gerdes, "Neural network model predictive motion control applied to automated driving with unknown friction," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 5, pp. 1934–1945, 2021.

[22] K. Y. Chee, T. Z. Jiahao, and M. A. Hsieh, "Knode-mpc: A knowledge-based data-driven predictive control framework for aerial robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2819–2826, 2022.

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

[24] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, "System level synthesis," *Annual Reviews in Control*, vol. 47, pp. 364–393, 2019.

[25] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[26] J. Köhler, E. Andina, R. Soloperto, M. A. Müller, and F. Allgöwer, "Linear robust adaptive model predictive control: Computational complexity and conservatism," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1383–1388, IEEE, 2019.

[27] J. Fleming, B. Kouvaritakis, and M. Cannon, "Robust tube mpc for linear systems with multiplicative uncertainty," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 1087–1092, 2014.

[28] M. Bujarbaruah, U. Rosolia, Y. R. Stürz, X. Zhang, and F. Borrelli, "Robust mpc for lpv systems via a novel optimization-based constraint tightening," *Automatica*, vol. 143, p. 110459, 2022.

[29] S. Chen, K. Y. Chee, N. Matni, M. A. Hsieh, and G. J. Pappas, "Safety filter design for neural network systems via convex optimization," *arXiv preprint arXiv:2308.08086*, 2023.

[30] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[31] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems.," in *ICINCO (1)*, pp. 222–229, Citeseer, 2004.

[32] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1168–1175, IEEE, 2014.

[33] B. Amos, I. Jimenez, J. Sacks, B. Boots, and J. Z. Kolter, "Differentiable mpc for end-to-end planning and control," *Advances in Neural Information Processing Systems*, vol. 31, 2018.