2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Approximations of Optimal Control Laws for Constrained Piecewise Affine Systems by Deep Neural Networks

Lukas Markolf, Arne Siewert, and Olaf Stursberg

Abstract— **The paper on hand considers the optimal control of piecewise affine systems subject to polytopic constraints. While this problem can be addressed by receding horizon control, the approach is known to be computationally demanding. This paper considers the approximation of receding horizon control laws by deep artificial feed-forward neural networks. The concept of projecting inadmissible inputs onto regions derived from feasible sets is extended to the considered problem setup in order to achieve deterministic guarantees on feasibility and constraint satisfaction. Two approaches are proposed and illustrated in numerical examples.**

I. Introduction

The optimal control of piecewise affine (PWA) systems has been studied intensively, motivated by the ability of PWA systems to approximate nonlinear systems and their equivalence to many classes of hybrid systems [1], [2]. Receding horizon control (RHC) is an established approach: at each sample time, the optimal input is either computed by a mixed-integer optimization, or recovered from a lookup table determined offline.

The paper on hand focuses on the often-considered class of discrete-time, time-invariant, and finite-dimensional PWA systems subject to polytopic input and state constraints. For this non-smooth class of nonlinear systems, the optimization problems solved by RHC may be formulated as mixed integer linear programs (MILPs) or mixed integer quadratic programs (MIQPs), given that the performance criteria are based on linear or quadratic norms, respectively [3]. However, the computational demand for repeated online solutions can be prohibitive due to the combinatorial nature of these problems. With suitable assumptions, explicit solutions in the form of time-varying PWA state-feedback control laws exist. Mayne sketched a proof of this result in his plenary presentation at the 2001 European Control Conference [4]. A detailed exposition of the proof can be found in [5], including a description of how the explicit PWA control laws can be constructed by combining multiparametric and dynamic programming. But even for the simple subclass of linear time invariant (LTI) systems, the offline determination of the explicit RHC laws becomes computationally intractable rapidly [6], and the demands for storage and online computation often get excessive [7].

Recently, the approximation of RHC laws by deep feed-forward neural networks (DNNs) has gained increased attention. In addition to their universal approximation property, DNNs can be stored and evaluated efficiently on low-cost embedded systems [8]. However, DNNs are characterized by a highly nonlinear structure, making it hard to provide closed-loop guarantees even for simple LTI systems. One approach for guaranteed constraint satisfaction is to project inputs onto (state-dependent) sets of admissible inputs derived from control invariant sets. This approach has been proposed for LTI systems in [9] and extended to the class of switched linear systems with externally forced switching in [10], [11]. The approximation of RHC laws for PWA systems by DNNs with rectified linear units (ReLUs) as activation functions is considered in [12]. In that work, the authors suggest an approach for learning Lyapunov functions, which may be used to find inner approximations of the region of attraction (ROA). In order to address hard constraints on states and inputs, they further discuss an extension of the method to projected DNN controllers for the particular subclass of LTI systems. A crucial requirement for the extension is that the projection relies on the solution of a single quadratic program (QP).

The paper on hand studies the extension of the projection approach to the class of PWA systems (in which the dynamics switch autonomously, which is different from the externally forced switching in [11]). It is shown that the state-dependent sets of admissible inputs derived from feasible state sets are, in general, non-convex unions of polytopes, which allows one to realize the projections by solving multiple QPs. Polytopic feasible state sets constitute an exception, in which case the projection requires only the solution of a single QP. Motivated by the ambition to reduce the computational demand, a method for computing polytopic feasible sets is proposed.

The paper is organized such that Sec. II covers the preliminaries and the problem formulation. The approximation of RHC laws by DNNs with deterministic guarantees on feasibility and constraint satisfaction is treated in Sec. III. Numerical examples are considered in Sec. IV, before conclusions are provided in Sec. V.

II. Preliminaries and Problem Formulation

This work considers PWA systems described by:

$$x_{k+1} = f_i(x_k, u_k) \text{ if } x_k \in \mathcal{P}_i, \quad i \in \mathcal{I}, \tag{1}$$

where $\mathcal{I} = \{1, \ldots, n_{\mathcal{I}}\}$ is a finite index set, $\{\mathcal{P}_i\}_{i\in\mathcal{I}}$ a polyhedral partition of the state space $\mathcal{X} \subset \mathbb{R}^{n_x}$, $x_k \in \mathcal{X}$ the state at time $k \in \mathbb{N}_0$, and $u_k \in \mathcal{U}$ the input selected from

L. Markolf, A. Siewert, and O. Stursberg are with the Inst. of Control and System Theory, Dept. of Electrical Eng. and Computer Science, University of Kassel, Germany, {lukas.markolf,stursberg}@uni-kassel.de

$\mathcal{U} \subset \mathbb{R}^{n_u}$. The function f_i denotes the state transition map in mode $i \in \mathcal{I}$ defined by A_i, B_i, and c_i:

$$f_i(x, u) = A_i x + B_i u + c_i. \tag{2}$$

The sets $\mathcal{U}$ and $\mathcal{X}$ are bounded polytopes with the origin in their interior, and are defined by vectors $h^{\mathcal{U}}$, $h^{\mathcal{X}}$ and matrices $H^{\mathcal{U}}$, $H^{\mathcal{X}}$ of appropriate dimensions:

$$\mathcal{U} = \left\{u \mid H^{\mathcal{U}} u \leq h^{\mathcal{U}}\right\}, \quad \mathcal{X} = \left\{x \mid H^{\mathcal{X}} x \leq h^{\mathcal{X}}\right\}. \tag{3}$$

Assumption 1. The PWA systems have the property that for each $x \in \mathcal{X}$ there exists only one $i \in \mathcal{I}$ for which $x \in \mathcal{P}_i$.

Remark 1. Assumption 1 is common in the literature and ensures the uniqueness of the state sequence. From a practical point of view, the property can be achieved by introducing gaps with magnitudes close to machine precision between the boundaries of any two polyhedra, as discussed in more detail in [3].

The following algorithm formulates a typical implementation of RHC with weighting matrices $P \succeq 0$, $Q \succeq 0$, and $R \succ 0$, a finite time horizon $N \in \mathbb{N}_0$, a predicted input sequence $\boldsymbol{u}_k = \{u_{0|k}, \ldots, u_{N-1|k}\}$, a cost function:

$$J_N(x_k, \boldsymbol{u}_k) = \|x_{N|k}\|_P^2 + \sum_{j=0}^{N-1} \|x_{j|k}\|_Q^2 + \|u_{j|k}\|_R^2, \tag{4}$$

and a polytopic terminal region $\mathcal{X}_f \subseteq \mathcal{X}$.

Algorithm 1 (RHC)**.** At each time instant $k \in \mathbb{N}_0$:

(i) Solve the optimization:

$$\begin{aligned} \underset{\boldsymbol{u}_k}{\text{minimize}} \quad & J_N(x_k, \boldsymbol{u}_k) \\ \text{subject to} \quad & x_{j+1|k} = f_i(x_{j|k}, u_{j|k}) \text{ if } x_{j|k} \in \mathcal{P}_i, \\ & x_{0|k} = x_k, \quad x_{N|k} \in \mathcal{X}_f, \\ & x_{j|k} \in \mathcal{X}, \quad u_{j|k} \in \mathcal{U}. \end{aligned} \tag{5}$$

(ii) Apply the control input $u_k = u^*_{0|k}$ from $\boldsymbol{u}^*_k = \{u^*_{0|k}, \ldots, u^*_{N-1|k}\}$ as obtained from solving (5).

The set of states for which a feasible solution of (5) exists is denoted by $\mathcal{X}_0$ and is recursively defined:

$$\begin{aligned} \mathcal{X}_j = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U}, \exists i \in \mathcal{I} \text{ such that} \\ x \in \mathcal{P}_i \text{ and } f_i(x, u) \in \mathcal{X}_{j+1}\}, \\ \mathcal{X}_N = \mathcal{X}_f, \quad j \in \{N-1, \ldots, 0\}. \end{aligned} \tag{6}$$

The terminal set $\mathcal{X}_f$ and the terminal weighting matrix P are typically chosen to guarantee asymptotic stability of the origin, e.g., by requiring the origin to be a cost-free state, $\mathcal{X}_f$ to be a control-invariant set, $0 \in \mathcal{X}_f$, and $p(x) = \|x\|_P^2$ to be a control Lyapunov function on $\mathcal{X}_f$.

Assumption 2. The terminal set $\mathcal{X}_f$ is control-invariant, i.e.:

$$x \in \mathcal{X}_f \Rightarrow \exists u \in \mathcal{U} \text{ such that } f_i(x, u) \in \mathcal{X}_f \text{ if } x \in \mathcal{P}_i.$$

Remark 2. If $\mathcal{X}_N = \mathcal{X}_f$ is control-invariant, then $\mathcal{X}_{j+1} \subseteq \mathcal{X}_j$. This implies that $\mathcal{X}_0$ is also control-invariant and recursive feasibility of Alg. 1 is ensured for each $x_0 \in \mathcal{X}_0$. On the other hand, the RHC law is infeasible for $x_k \notin \mathcal{X}_0$. In this case, $x_{0|k} = x_k$ cannot be transferred into the terminal set within N steps, such that the constraint $x_{N|k} \in \mathcal{X}_f$ cannot be met. Consequently, an equivalent problem of (5) is obtained when replacing the constraint $x_{j|k} \in \mathcal{X}$ by $x_{j|k} \in \mathcal{X}_0$, in which case the RHC law remains unchanged (in Sec. III-B, the substitution of $x_{j|k} \in \mathcal{X}$ by $x_{j|k} \in \tilde{\mathcal{X}}_0 \subseteq \mathcal{X}_0$ is considered, leading to a potentially suboptimal RHC law). The set $\mathcal{X}_0$ grows with increasing N until it becomes the maximal stabilizable set within $\mathcal{X}$.

The PWA systems considered here can be rewritten as mixed logical dynamical (MLD) systems [2]. MLD representations of (1) then allow to reformulate (5) as an MIQP [13]. The RHC law from Alg. 1 is a PWA state feedback control law $u_k = \mu_{\text{RHC}}(x_k)$ defined on possibly non-convex regions.

This work uses the perspective that the online application of RHC based on (i) the solution of an MIQP or (ii) the evaluation of the explicit control law (by recovering the input from a stored lookup table) is computationally intractable. It is further supposed that the available computational resources allow the offline generation of enough state-input pairs to train a suitably chosen parametric function by supervised learning to approximate the RHC law with satisfactory quality. DNNs are used as parametric functions:

$$h(z; \theta) = (h_L \circ h_{L-1} \circ \ldots \circ h_1)(z), \tag{7}$$

$$h_l(\eta_{l-1}) = \phi_l(W_l \eta_{l-1} + b_l), \quad \eta_0 = z, \tag{8}$$

with input $z \in \mathbb{R}^{s_0}$, layers $l \in \{1, \ldots, L\}$, activation functions $\phi_l : \mathbb{R}^{s_{l-1}} \to \mathbb{R}^{s_l}$, and a parameter vector $\theta \in \mathbb{R}^{n_\theta}$ consisting of the elements of the weight matrices W_l and bias vectors b_l. The approximated RHC law is written as:

$$u_k = \mu_{\text{DNN}}(x_k) = h(x_k; \theta). \tag{9}$$

Due to the large and highly nonlinear structure of DNNs, it is challenging to provide guarantees on feasibility and constraint satisfaction. Addressing this challenge is the very objective of this work.

III. PROJECTION TO FEASIBLE INPUTS

For a given state $x_k \in \mathcal{P}_i$ at time k, the set of inputs $u_k \in \mathcal{U}$ for which $x_{k+1} = f_i(x_k, u_k)$ is an element of $\mathcal{X}_0$ is given as $\underline{\mathcal{U}}(x_k)$ with:

$$\underline{\mathcal{U}}(x) = \{u \in \mathcal{U} \mid f_i(x, u) \in \mathcal{X}_0 \text{ if } x \in \mathcal{P}_i\}. \tag{10}$$

The property of control-invariance of $\mathcal{X}_0$ implies that $\underline{\mathcal{U}}(x)$ is nonempty for all $x \in \mathcal{X}_0$. The following proposition follows directly from the definitions above.

Proposition 1. The system (1) with $x_0 \in \mathcal{X}_0$ and the controller μ_{DNN} satisfy the constraints $x_k \in \mathcal{X}$ and $u_k \in \mathcal{U}$ if $\mu_{\text{DNN}}(x_k) \in \underline{\mathcal{U}}(x_k)$ for all $k \in \mathbb{N}_0$.

In general, it is challenging to verify that $\mu_{\text{DNN}}(x) \in \underline{\mathcal{U}}(x)$ for all $x \in \mathcal{X}_0$. Furthermore, it may be difficult to find a parameter vector θ at all for which the condition above is met. The problem gets even more involved if θ is determined online (which is not considered in this paper).

Let $u_{\text{DNN},k} = \mu_{\text{DNN}}(x_k)$ be the output of the DNN controller for state x_k at time k. To obtain an input that is guaranteed to be feasible and satisfies the constraints, the following optimization is solved:

$$\begin{aligned} \underset{u_{\text{PROJ},k}}{\text{minimize}} \quad & \|u_{\text{PROJ},k} - u_{\text{DNN},k}\|_2 \\ \text{subject to} \quad & u_{\text{PROJ},k} \in \underline{\mathcal{U}}(x_k). \end{aligned} \tag{11}$$

Its solution selects u_k to be an element of $\underline{\mathcal{U}}(x_k)$ with minimum Euclidean distance to $u_{\text{DNN},k}$, i.e., the latter is projected onto the admissible input set. For $u_{\text{DNN},k} \in \underline{\mathcal{U}}(x_k)$, of course $u_k = u_{\text{DNN},k}$ applies.

Remark 3. The projection defined in (11) is feasible if and only if $\underline{\mathcal{U}}(x_k)$ is nonempty, which is given for all $x_k \in \mathcal{X}_0$. There may also be $x \notin \mathcal{X}_0$ for which $\underline{\mathcal{U}}(x)$ is nonempty, as long as $\mathcal{X}_0$ is not the maximal stabilizable set in $\mathcal{X}$. On the contrary, the RHC law is only feasible on $\mathcal{X}_0$, such that the domain of the RHC law is a subset of the domain of the projected DNN controller.

Remark 4. The RHC law defined in Alg. 1 has the property that $\mu_{\text{RHC}}(x) \in \underline{\mathcal{U}}(x)$ for all $x \in \mathcal{X}_0$, i.e., the capability of the DNN to approximate the RHC law is not affected by the projection (in other words, the output of a DNN is not changed by the projection if it approximates the RHC law with zero approximation error).

The following subsection intends to show that $\underline{\mathcal{U}}(x)$ is the union of polytopes for each state $x \in \mathcal{X}_0$ and that the projection can be established by the solution of (in general multiple) QPs. Sec. III-B is going to describe an approach to compute polytopic control invariant sets $\tilde{\mathcal{X}}_0 \subseteq \mathcal{X}_0$, in which case the projection relies only on the solution of a single QP. The method proposed for computing the sets $\tilde{\mathcal{X}}_0$ is inspired by the one suggested for switched systems with externally forced switching in [11].

A. Approximate RHC with non-convex feasible state set

The feasible set $\mathcal{X}_0$ is the union of (possibly overlapping) polytopes and thus forms a non-convex set (see, e.g., [14]). Subsequently, let $\cup_{j\in\mathcal{J}}\mathcal{F}_j$ be a polytopic partition of the feasible set $\mathcal{X}_0$ with finite index set $\mathcal{J} = \{1, \dots, n_{\mathcal{J}}\}$, i.e.:

$$\mathcal{X}_0 = \cup_{j\in\mathcal{J}}\mathcal{F}_j, \tag{12}$$

$$\mathcal{F}_j = \left\{x \mid H^{\mathcal{F}_j} x \leq h^{\mathcal{F}_j}\right\}. \tag{13}$$

Theorem 1. Given Assumption 2, the state-dependent set $\underline{\mathcal{U}}(x)$ of feasible inputs defined in (10) is the nonempty union of polytopes for each $x \in \mathcal{X}_0$.

Proof. By inserting $\mathcal{X}_0 = \cup_{j\in\mathcal{J}}\mathcal{F}_j$ into (10), the set $\underline{\mathcal{U}}(x)$ can be reformulated as the union:

$$\underline{\mathcal{U}}(x) = \cup_{j\in\mathcal{J}}\underline{\mathcal{U}}_j(x), \tag{14}$$

with $\underline{\mathcal{U}}_j(x)$ defined to:

$$\underline{\mathcal{U}}_j(x) = \{u \in \mathcal{U} \mid f_i(x,u) \in \mathcal{F}_j \text{ if } x \in \mathcal{P}_i\}. \tag{15}$$

The sets $\underline{\mathcal{U}}_j(x)$ (depending on f_i according to (2) and $\mathcal{F}_j$ according to (13)) can be rewritten into:

$$\begin{aligned} \underline{\mathcal{U}}_j(x) = \{u \in \mathcal{U} \mid & H^{\mathcal{F}_j} B_i u \leq \\ & h^{\mathcal{F}_j} - H^{\mathcal{F}_j}(A_i x + c_i) \text{ if } x \in \mathcal{P}_i\}. \end{aligned} \tag{16}$$

By use of the half-space representation of $\mathcal{U}$ one eventually obtains:

$$\underline{\mathcal{U}}_j(x) = \{u \mid H_i^{\mathcal{U}_j} u \leq h_i^{\mathcal{U}_j}(x) \text{ if } x \in \mathcal{P}_i\}, \tag{17}$$

with matrices:

$$H_i^{\mathcal{U}_j} = \begin{bmatrix} H^{\mathcal{F}_j} B_i \\ H^{\mathcal{U}} \end{bmatrix}, \quad h_i^{\mathcal{U}_j}(x) = \begin{bmatrix} h^{\mathcal{F}_j} - H^{\mathcal{F}_j}(A_i x + c_i) \\ h^{\mathcal{U}} \end{bmatrix}.$$

Since $\mathcal{U}$ was defined polytopic, each nonempty set $\underline{\mathcal{U}}_j(x)$ is polytopic, too. The fact that at least one $\underline{\mathcal{U}}_j(x)$ is nonempty follows from the nonemptiness of $\underline{\mathcal{U}}(x)$ for each $x \in \mathcal{X}_0$ (which is a consequence of the control invariance of $\mathcal{X}_0$ given by Assumption 2). □

For a given $x \in \mathcal{X}$, let the set of indices $j \in \mathcal{J}$ with nonempty $\underline{\mathcal{U}}_j(x)$ be denoted by:

$$\underline{\mathcal{J}}(x) = \left\{j \in \mathcal{J} \mid \underline{\mathcal{U}}_j(x) \neq \emptyset\right\}. \tag{18}$$

The set $\underline{\mathcal{J}}(x)$ can be determined by solving the convex feasibility problem:

$$\text{find} \quad u \quad \text{subject to} \quad u \in \underline{\mathcal{U}}_j(x) \tag{19}$$

for each $j \in \mathcal{J}$. It follows from the proof of Theorem 1 that $\underline{\mathcal{J}}(x)$ is nonempty for all $x \in \mathcal{X}_0$.

Based on these aspects, the approximation of the RHC law from Alg. 1 by a DNN controller with guaranteed feasibility and constraint satisfaction is established by the following algorithm.

Algorithm 2 (Approximate RHC)**.** For each $k \in \mathbb{N}_0$:

(i) Compute $u_{\text{DNN},k} = \mu_{\text{DNN}}(x_k)$ from (9) and determine $\underline{\mathcal{J}}(x_k)$ by solving (19) for each $j \in \mathcal{J}$.

(ii) Solve:

$$\begin{aligned} \underset{u_{\text{PROJ},k}}{\text{minimize}} \quad & \|u_{\text{PROJ},k} - u_{\text{DNN},k}\|_2^2 \\ \text{subject to} \quad & u_{\text{PROJ},k} \in \underline{\mathcal{U}}_j(x_k), \quad j \in \underline{\mathcal{J}}(x_k). \end{aligned} \tag{20}$$

(iii) Apply the solution $u_k = u^*_{\text{PROJ},k}$ of (20).

Remark 5. Problem (20) is equivalent to problem (11); its solution can be obtained by first solving the QP:

$$\begin{aligned} \underset{u^{(j)}_{\text{PROJ},k}}{\text{minimize}} \quad & \|u^{(j)}_{\text{PROJ},k} - u_{\text{DNN},k}\|_2^2 \\ \text{subject to} \quad & u^{(j)}_{\text{PROJ},k} \in \underline{\mathcal{U}}_j(x_k) \end{aligned} \tag{21}$$

for each $j \in \underline{\mathcal{J}}(x_k)$ and then selecting from the resulting set of solutions $\mathcal{S}_k = \{u^{(j)^*}_{\text{PROJ},k}\}_{j\in\underline{\mathcal{J}}(x_k)}$ an element with minimum Euclidean norm to the controller output:

$$u^*_{\text{PROJ},k} \in \underset{u\in\mathcal{S}_k}{\arg\min} \|u - u_{\text{DNN},k}\|_2^2. \tag{22}$$

Remark 6. The computational demand of Alg. 2 grows with the number $n_{\mathcal{J}}$ of indices j in $\mathcal{J}$. Hence, a polytopic partition $\cup_{j\in\mathcal{J}}\mathcal{F}_j$ of a non-convex $\mathcal{X}_0$ with the least possible $n_\mathcal{J}$ is desirable. The problem of finding such a minimal representation by merging polytopes is studied, e.g., in [15]. Example 1 shows that $n_\mathcal{J}$ can be significantly smaller than the regions over which the explicit RHC law is defined (the offline determination of the latter becomes computationally intractable rapidly with increasing N).

The computation of $\mathcal{X}_0$ gets more demanding with increasing problem size, as does the determination of the least possible $n_\mathcal{J}$. Thus, the following subsection presents a computationally less demanding procedure for computing a polytopic control-invariant subset $\tilde{\mathcal{X}}_0$ of a non-convex $\mathcal{X}_0$, with the property that the RHC law is recursively feasible when substituting the constraint $x_{j|k}\in\mathcal{X}$ in (5) by $x_{j|k}\in\tilde{\mathcal{X}}_0$.

B. Approximate RHC with convex feasible state set

The paper on hand proposes the computation of polytopic sets $\tilde{\mathcal{X}}_0\subseteq\mathcal{X}_0$ by the recursion:

$$\begin{aligned}\tilde{\mathcal{X}}_j &= \mathcal{X}\cap\left(\bigcap_{i\in\mathcal{I}}\mathrm{Pre}_i(\tilde{\mathcal{X}}_{j+1})\right),\\ \tilde{\mathcal{X}}_N &= \mathcal{X}_f,\quad j\in\{0,\dots,N-1\},\end{aligned}\tag{23}$$

where $\mathrm{Pre}_i(\mathcal{S})$ denotes the set of predecessors of states in $\mathcal{S}\subseteq\mathcal{X}$ according to:

$$\mathrm{Pre}_i(\mathcal{S})=\{x\mid\exists u\in\mathcal{U}\text{ such that }f_i(x,u)\in\mathcal{S}\}\,.\tag{24}$$

The following assumption ensures control-invariance of $\tilde{\mathcal{X}}_0$.

Assumption 3. The polytopic terminal set $\mathcal{X}_f\subseteq\mathcal{X}$ is control-invariant for all dynamics $x_{k+1}=f_i(x_k,u_k)$:

$$x\in\mathcal{X}_f\;\Rightarrow\;\forall i\in\mathcal{I}:\exists u\in\mathcal{U}\text{ such that }f_i(x,u)\in\mathcal{X}_f.$$

Theorem 2. Given Assumption 3, the state-dependent set

$$\underline{\tilde{\mathcal{U}}}(x)=\left\{u\in\mathcal{U}\mid f_i(x,u)\in\tilde{\mathcal{X}}_0\text{ if }x\in\mathcal{P}_i\right\}\tag{25}$$

is a nonempty polytopic subset of $\underline{\mathcal{U}}(x)$ for all $x\in\tilde{\mathcal{X}}_0$.

Proof. The set $\mathrm{Pre}_i(\mathcal{X}_f)$ consists of all states $x\in\mathcal{X}$ for which at least one $u\in\mathcal{U}$ can be selected such that $f_i(x,u)\in\mathcal{X}_f$. Hence, $\mathcal{X}_f\subseteq\mathrm{Pre}_i(\mathcal{X}_f)$ applies for all $i\in\mathcal{I}$ according to Assumption 3. Thus, $\mathcal{X}_f\subseteq\cap_{i\in\mathcal{I}}\mathrm{Pre}_i(\tilde{\mathcal{X}}_{j+1})$ follows (this property would not be given if $\mathcal{X}_f\not\subseteq\mathrm{Pre}_i(\mathcal{X}_f)$ for at least one $i\in\mathcal{I}$). Moreover, $\mathcal{X}_f\subseteq\mathcal{X}$ and $\tilde{\mathcal{X}}_N=\mathcal{X}_f$ holds, such that $\tilde{\mathcal{X}}_N\subseteq\tilde{\mathcal{X}}_{N-1}=\mathcal{X}\cap(\cap_{i\in\mathcal{I}}\mathrm{Pre}_i(\tilde{\mathcal{X}}_N))$. Consequently, for each $x\in\tilde{\mathcal{X}}_{N-1}$ and $i\in\mathcal{I}$, there exists at least one u such that $f_i(x,u)\in\tilde{\mathcal{X}}_N\subseteq\tilde{\mathcal{X}}_{N-1}$, i.e. $\tilde{\mathcal{X}}_{N-1}$ is, just as $\tilde{\mathcal{X}}_N$, control-invariant for all dynamics $x_{k+1}=f_i(x_k,u_k)$. The property that $\tilde{\mathcal{X}}_{j+1}\subseteq\tilde{\mathcal{X}}_j$ for each $j\in\{0,1,\dots,N-1\}$ and the control-invariance of $\tilde{\mathcal{X}}_{j+1}$ for all dynamics $x_{k+1}=f_i(x_k,u_k)$ follows by induction.

The nonemptiness of $\tilde{\mathcal{X}}_N=\mathcal{X}_f$ implies the nonemptiness of $\tilde{\mathcal{X}}_0\supseteq\tilde{\mathcal{X}}_1\supseteq\ldots\supseteq\tilde{\mathcal{X}}_N$. Hence, $\underline{\tilde{\mathcal{U}}}(x)$ is nonempty for all $x\in\tilde{\mathcal{X}}_0$. The sets $\mathrm{Pre}_i(\mathcal{S})$ are the result of linear operations on $\mathcal{S}$ and $\mathcal{U}$: if $\mathcal{S}$ is a polytope, then $\mathrm{Pre}_i(\mathcal{S})$ is a polytope, and the assumption of a polytopic terminal set $\mathcal{X}_f$ implies that all $\tilde{\mathcal{X}}_j$ are polytopic:

$$\tilde{\mathcal{X}}_j=\left\{x\mid H^{\tilde{\mathcal{X}}_j}x\leq h^{\tilde{\mathcal{X}}_j}\right\},\quad j\in\{0,1,\dots,N\}.\tag{26}$$

By inserting the expression of the state transition map (2) and the half-space representation of $\tilde{\mathcal{X}}_0$ into (25), the set $\underline{\tilde{\mathcal{U}}}(x)$ can be rewritten as:

$$\underline{\tilde{\mathcal{U}}}(x)=\left\{u\mid H_i^{\underline{\tilde{\mathcal{U}}}}u\leq h_i^{\underline{\tilde{\mathcal{U}}}}(x)\text{ if }x\in\mathcal{P}_i\right\},\tag{27}$$

with

$$H_i^{\underline{\tilde{\mathcal{U}}}}=\begin{bmatrix}H^{\tilde{\mathcal{X}}_0}B_i\\ H^{\mathcal{U}}\end{bmatrix},\quad h_i^{\underline{\tilde{\mathcal{U}}}}(x)=\begin{bmatrix}h^{\tilde{\mathcal{X}}_0}-H^{\tilde{\mathcal{X}}_0}(A_ix+c_i)\\ h^{\mathcal{U}}\end{bmatrix}.$$

Hence, $\underline{\tilde{\mathcal{U}}}(x)$ is a nonempty polytope for all $x\in\tilde{\mathcal{X}}_0$.

The set $\mathcal{X}_0$, which is recursively defined in (6), consists of all states $x\in\mathcal{X}$ that can be transferred into $\mathcal{X}_f$ within N steps while satisfying the state and input constraints in (3). It follows directly from the recursive definition of $\tilde{\mathcal{X}}_0$ in (23) that for each $x_k\in\tilde{\mathcal{X}}_0$ there exists at least one input sequence $\{u_k,u_{k+1},\dots,u_{k+N-1}\}$ such that $u_{k+j}\in\mathcal{U}$ and $x_{k+j+1}\in\tilde{\mathcal{X}}_{j+1}$ for all $j\in\{0,1,\dots,N-1\}$. Thus, $\tilde{\mathcal{X}}_0\subseteq\mathcal{X}_0$, and it finally follows from the definition of $\underline{\mathcal{U}}(x)$ in (10) that $\underline{\tilde{\mathcal{U}}}(x)\subseteq\underline{\mathcal{U}}(x)$ for all $x\in\tilde{\mathcal{X}}_0$. □

By restricting the set of admissible states to $\tilde{\mathcal{X}}_0$, a simplified algorithm can be formulated for approximating the RHC law, see Algorithm 3 below.

Algorithm 3 (Approximate RHC with convex feasible set)**.** At each time instant $k\in\mathbb{N}_0$:

(i) Compute $u_{\mathrm{DNN},k}=\mu_{\mathrm{DNN}}(x_k)$ by evaluating the DNN control law (9).

(ii) Solve:

$$\begin{aligned}\underset{u_{\mathrm{PROJ},k}}{\text{minimize}}\quad&\|u_{\mathrm{PROJ},k}-u_{\mathrm{DNN},k}\|_2^2\\ \text{subject to}\quad&u_{\mathrm{PROJ},k}\in\underline{\tilde{\mathcal{U}}}(x_k).\end{aligned}\tag{28}$$

(iii) Apply the solution $u_k=u^*_{\mathrm{PROJ},k}$ of (28).

Remark 7. Algorithm 3 requires only the solution of a single QP for projection at the cost of potential suboptimality due to the property that $\underline{\tilde{\mathcal{U}}}(x)\subseteq\underline{\mathcal{U}}(x)$ for $x\in\tilde{\mathcal{X}}_0$. It is advisable to approximate RHC laws for which the constraint $x_{j|k}\in\mathcal{X}$ in (5) is substituted by $x_{j|k}\in\tilde{\mathcal{X}}_0$. Subsequently, these RHC laws are denoted as suboptimal RHC laws. On the one hand, the suboptimal RHC laws have the property that $\mu_{\mathrm{RHC}}(x)\in\underline{\tilde{\mathcal{U}}}(x)$ for all $x\in\tilde{\mathcal{X}}_0$, i.e., the capability of the DNNs to approximate the suboptimal RHC laws is not affected by the projection. On the other hand, recursive feasibility of the suboptimal RHC laws is ensured for each $x_0\in\tilde{\mathcal{X}}_0$, as follows directly from the proof of Theorem 2. Since the projection ensures that $x_0\in\tilde{\mathcal{X}}_0$ implies $x_k\in\tilde{\mathcal{X}}_0$ for all $k\in\mathbb{N}_0$, there is always an input sequence $\{u_k,u_{k+1},\dots,u_{k+N-1}\}$ which transfers x_k into the terminal region $\mathcal{X}_f$ while satisfying the constraints in (3).

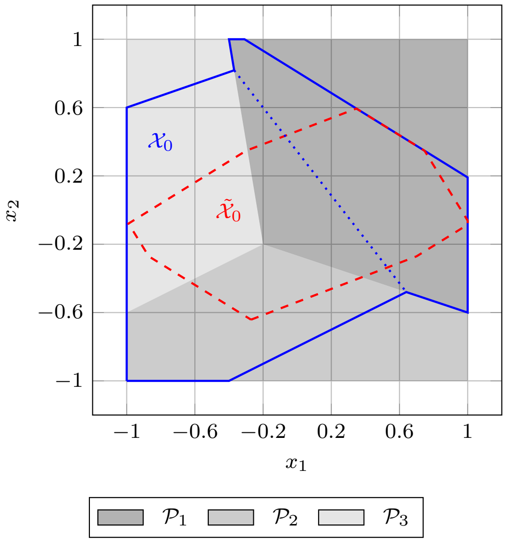

Fig. 1. Example 1: Shaded areas illustrate the polyhedral partition $\{\mathcal{P}_i\}_{i\in\{1,2,3\}}$ of $\mathcal{X}$. Solid blue lines represent the boundary of $\mathcal{X}_0$ and dashed red lines the boundary of $\tilde{\mathcal{X}}_0$. The blue dotted line indicates that $\mathcal{X}_0$ can be partitioned into two polytopes.

IV. SIMULATION EXAMPLE

A. *Example* 1 - 2*D System*

Consider a PWA system (1) that is parameterized by:

$$A_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 1 \\ -0.5 & 1 \end{bmatrix},$$
$$B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, c_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, c_2 = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}, c_3 = \begin{bmatrix} 0 \\ -0.2 \end{bmatrix}$$

(i.e., $B_1 = B_2 = B_3 = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$) and is subject to:

$$x_k \in \mathcal{X} = \{x \mid |x_i| \le 1\}, \quad u_k \in \mathcal{U} = \{u \mid |u| \le 0.5\},$$

with polyhedral partition $\{\mathcal{P}_i\}_{i\in\{1,2,3\}}$ of $\mathcal{X}$ as visualized in Fig. 1 (some of the inequalities defining $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$ are rendered to strict form to ensure compliance with Assumption 1). Further, consider an RHC setup with $N = 20$, $\mathcal{X}_f = \{0_{2\times 1}\}$, $P = 0_{2\times 2}$, $Q = I_{2\times 2}$, and $R = 1$. The sets $\mathcal{X}_0$ and $\tilde{\mathcal{X}}_0$ obtained for the chosen N are shown in Fig. 1 as well, with a blue dotted line indicating the subsequently considered partition of $\mathcal{X}_0$ into polytopes $\mathcal{F}_1$ and $\mathcal{F}_2$.

The approximation of the RHC law by a DNN controller μ_{DNN} according to Alg. 2 was carried out for $L = 6$ (number of layers) and $s_l = 30$, $l \in \{1, \dots, L-1\}$ (number of hidden units). As activation functions, ReLUs were used, except for the last layer, for which the identity function was chosen. The DNN controller was trained offline by state-input pairs $\{(x^{(q)}, u^{(q)})\}_{q\in\mathcal{Q}}$, $\mathcal{Q} = \{1, \dots, 10^4\}$; the states $x^{(q)}$ were determined by gridding $\mathcal{X}_0$, while the inputs $u^{(q)} = \mu_{\text{RHC}}(x^{(q)})$ were obtained by evaluating the RHC law from Alg. 1. For these evaluations, problem (5) has been reformulated and solved as MIQP. The offline computation of an explicit RHC law was found to be intractable due to storage requirements. Fig. 2 compares the state sequences generated by Alg. 1 (left plot) and Alg. 2 (middle plot)

TABLE I

EXPLICIT RHC: GROWTH IN THE NUMBER OF REGIONS WITH INCREASING PREDICTION HORIZON.

N	2	3	4	5	6
n_r	6	39	184	663	2135

for $x_0 = \begin{bmatrix} -1 & -1 \end{bmatrix}^\top$. Further, it compares the values of $\mu_{\text{RHC}}(x(\lambda))$ and $\mu_{\text{DNN}}(x(\lambda))$ (right plot) along the line segment from $x^{(1)} = \begin{bmatrix} -1 & -0.4 \end{bmatrix}^\top$ to $x^{(2)} = \begin{bmatrix} 1 & -0.4 \end{bmatrix}^\top$:

$$x(\lambda) = \lambda x^{(1)} + (1-\lambda)x^{(2)}, \quad 0 \le \lambda \le 1.$$

In this example, $\mathcal{X}_0$ is equal to the maximal stabilizable set in $\mathcal{X}$, i.e. there is no admissible control law that transfers $x_k \notin \mathcal{X}_0$ into $\mathcal{X}_f$ within a finite number of steps. Moreover, $\mathcal{X}_0$ remains the same when further increasing N, such that the partition of $\mathcal{X}_0$ into two polytopes $\mathcal{F}_1$ and $\mathcal{F}_2$ can be maintained. Consequently, the evaluation of Alg. 2 only requires, even for $N > 20$, the evaluation of a closed-form expression (the DNN) and the solution of at most two QPs for projection. On the contrary, the computational complexity for evaluating Alg. 1 grows with an increasing N. The rapid growth in the number n_r of regions of the explicit RHC law with increasing N is documented in Tab. I. The approximation of the suboptimal RHC law by Alg. 3 leads to comparably close approximation results. However, the solution of a single instead of at most two QPs does not cause a significant increase in computational efficiency in this particular example.

B. *Example* 2 - 3*D System*

Consider now a PWA system in $\mathbb{R}^3$ parameterized by:

$$A_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 1.8 & 1 & 1.6 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ -1.8 & 1 & 1.6 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1.4 \\ -1.4 & 2 & 0.6 \end{bmatrix}, A_4 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -2 \\ -1.4 & 0.8 & 0.6 \end{bmatrix},$$
$$B_i = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, c_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, c_2 = \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix}, c_{3/4} = \begin{bmatrix} 0 \\ -0.1 \\ 0.1 \end{bmatrix},$$

with states and inputs constrained to:

$$\mathcal{X} = \{x \mid |x_i| \le 2\}, \quad \mathcal{U} = \{u \mid |u| \le 1\},$$

and with a partition of $\mathcal{X}$ into:

$$\mathcal{P}_1 \subseteq \left\{ x \in \mathcal{X} \mid \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} \right\},$$
$$\mathcal{P}_2 \subseteq \left\{ x \in \mathcal{X} \mid \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x \le \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix} \right\},$$
$$\mathcal{P}_3 \subseteq \left\{ x \in \mathcal{X} \mid \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x \le \begin{bmatrix} -0.1 \\ -0.1 \end{bmatrix} \right\},$$
$$\mathcal{P}_4 \subseteq \left\{ x \in \mathcal{X} \mid \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} x \le \begin{bmatrix} -0.1 \\ 0.1 \end{bmatrix} \right\},$$

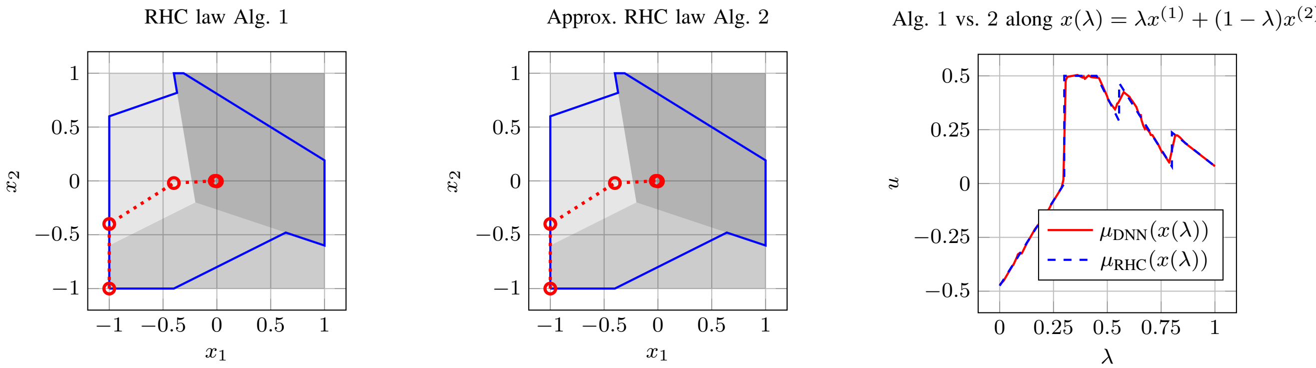

Fig. 2. Example 1: (Left) State sequence obtained from Alg. 1. (Middle) State sequence obtained from Alg. 2. (Right) Values $\mu_{\mathrm{RHC}}(x(\lambda))$ and $\mu_{\mathrm{DNN}}(x(\lambda))$ along the line segment $x(\lambda) = \lambda x^{(1)} + (1-\lambda)x^{(2)}$, $0 \leq \lambda \leq 1$ from $x^{(1)} = \begin{bmatrix} -1 & -0.4 \end{bmatrix}^\top$ to $x^{(2)} = \begin{bmatrix} 1 & -0.4 \end{bmatrix}^\top$.

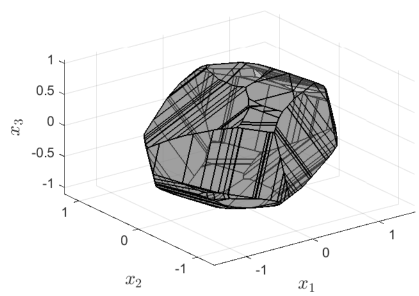

Fig. 3. Example 2: Illustration of the polytopic and feasible set $\tilde{\mathcal{X}}_0$.

TABLE II

EXAMPLE 2 - AVERAGE ONLINE COMPUTATION TIMES FOR EVALUATING THE LAW FROM ALG. 1 AND APPROXIMATION FROM ALG. 3

Control Law	Average Computation Times
Alg. 1 (YALMIP - MIQP)	1120 ms
Alg. 1 (Gurobi - MIQP)	93 ms
Alg. 1 (Explicit - MPT)	410 ms
Alg. 3 (DNN)	7 ms

where some of the inequalities defining $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, and $\mathcal{P}_4$ are rendered to strict form to ensure compliance with Assumption 1. The RHC setup is chosen with $N = 6$, $\mathcal{X}_f = \{0_{3\times 1}\}$, $P = 0_{3\times 3}$, $Q = I_{3\times 3}$, and $R = I_{2\times 2}$. The prediction horizon was chosen smaller than in the previous example to also allow the computation of the explicit RHC law.

The suboptimal RHC law was approximated by a DNN controller with the same structure as before, using Alg. 3. This time, the DNN controller was trained with state-input pairs $\{(x^{(q)}, u^{(q)})\}_{q\in\mathcal{Q}}$, $\mathcal{Q} = \{1, \ldots, 10^5\}$, obtained by gridding the set $\tilde{\mathcal{X}}_0$ shown in Fig. 3. Table II shows average online computation times for evaluating the RHC law from Alg. 1 and its approximation by Alg. 3 for $\{x^{(q)}\}_{q\in\mathcal{Q}}$ obtained from gridding $\tilde{\mathcal{X}}_0$. The computation times of Alg. 1 refer to (i) the evaluation based on the solutions of MIQPs and (ii) the evaluations of an explicit expression of the RHC law determined by the MPT toolbox [16]. The average evaluation time for the explicit expression is evidently faster than the one obtained with the non-commercial MIQP solver of YALMIP [17], but clearly slower than the one obtained with the commercial MIQP solver implemented in Gurobi [18]. On the other hand, the average computation time for evaluating Alg. 3 with the DNN controller is more than 10 times faster than the average computation time for evaluating Alg. 1 with the MIQP solver in Gurobi. Finally, Fig. 4 shows the state sequences generated by the RHC law and its approximation for a selected x_0.

V. CONCLUSION

This paper has addressed the approximation of RHC laws for PWA systems by DNNs, and deterministic guarantees on constraint satisfaction and feasibility have been obtained by projecting the outputs of the DNNs onto sets of inputs derived from feasible state sets. In contrast to LTI systems, the convexity of feasible input sets is, in general, not given for the broader class of PWA systems, to which the projection approach has (to the best of the author's knowledge) not been extended before.

The first contributions of the paper were to show 1) that the set of feasible inputs is obtained as the union of polytopes and 2) that the projection can be realized by solving several QPs whose number depends on the polytopic partition of the feasible state set. In contrast to the projected DNN controller, the demand for computing the RHC law grows with an increasing prediction horizon and becomes computationally intractable rapidly. Sec. IV-A provided an example for which the projection required only the solution of two QPs for an arbitrarily large prediction horizon. At the same time, the offline computation of the explicit RHC law was found to be intractable even for small prediction horizons.

Another contribution of the paper on hand was to present a computationally less demanding procedure for determining polytopic control-invariant subsets of feasible state sets, allowing the realization of the projection - at the cost of potential suboptimality - by only solving a single QP. The benefits

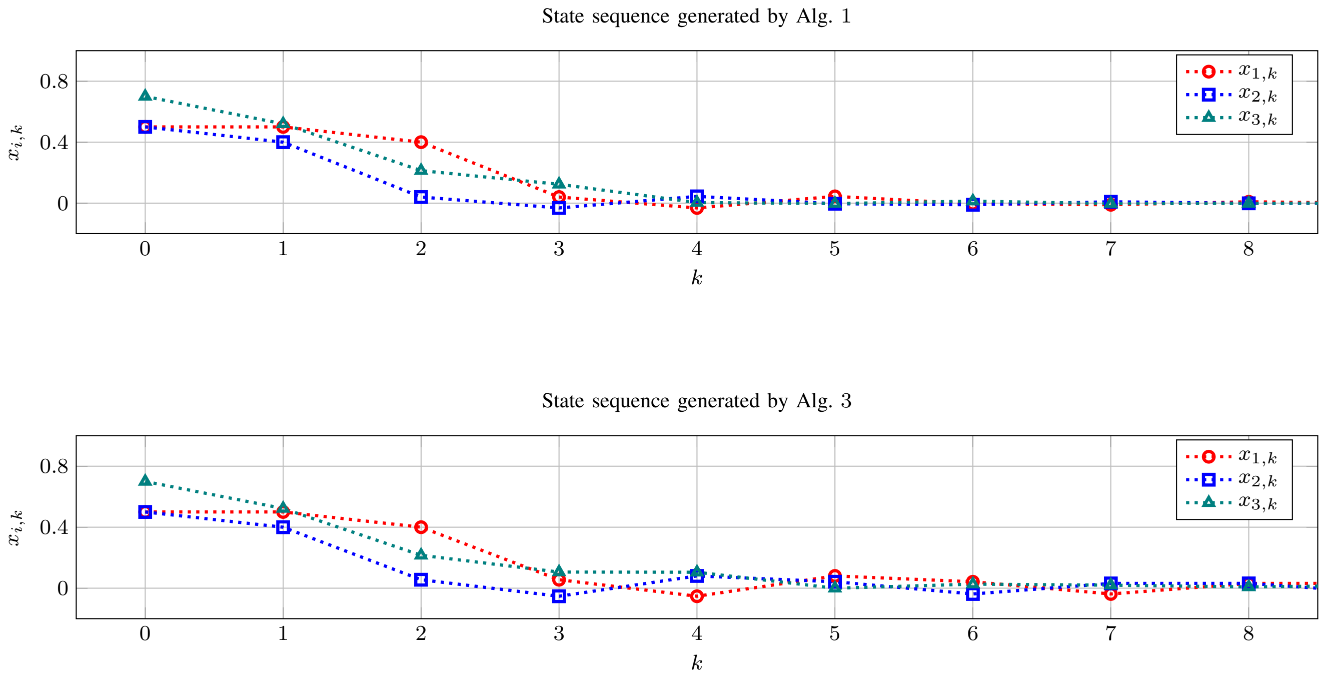

Fig. 4. Example 2: State sequences generated by the RHC law and its approximation for a selected initial state.

of this approach with respect to computational efficiency compared to RHC evaluations have been demonstrated even for small prediction horizons in Sec. IV-B.

The focus of this paper has been the guaranteed satisfaction of constraints rather than the search for the most suited DNN controllers. However, approximations of RHC laws with high accuracy have been achieved without large tuning effort in the simulation examples. Nevertheless, the discontinuity of the RHC laws may motivate considering other activation functions, network architectures, or classes of approximators in subsequent research. As a matter of fact, the projection approach can be used for any type of control law. A point not addressed here is the efficient generation of training data. Of course, the gridding of the state-space suffers from the curse of dimensionality, but there is ongoing research addressing this issue elsewhere, see e.g. [19]. The analysis of stability was also out of the scope of this paper. However, the proposed projection by a single QP seems to allow a straightforward extension of the approach in [12] from LTI to PWA systems to find inner approximations of ROAs for projected DNN controllers.

REFERENCES

[1] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Trans. Automat. Contr.*, vol. 26, no. 2, pp. 346–358, 1981.

[2] W. Heemels, B. de Schutter, and A. Bemporad, "On the equivalence of classes of hybrid dynamical models," in *Proceedings of the 40th IEEE Conference on Decision and Control (CDC)*, 2001, pp. 364–369.

[3] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[4] D. Q. Mayne, "Constrained optimal control," in *European Control Conference (ECC)*, 2001, Plenary Lecture.

[5] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," *Automatica*, vol. 41, no. 10, pp. 1709–1721, 2005.

[6] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear Model Predictive Control*, M. Morari, M. Thoma, L. Magni, D. M. Raimondo, and F. Allgöwer, Eds. Springer Berlin Heidelberg, 2009, vol. 384, pp. 345–369.

[7] B. Kouvaritakis and M. Cannon, *Model Predictive Control*. Springer International Publishing, 2016.

[8] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3866–3878, 2020.

[9] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *Proceedings of the 2018 Annual American Control Conference (ACC)*, 2018, pp. 1520–1527.

[10] L. Markolf and O. Stursberg, "Learning-based optimal control of constrained switched linear systems using neural networks," in *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics*, 2021, pp. 90–98.

[11] ——, "Receding-horizon control of constrained switched systems with neural networks as parametric function approximators," *SN Computer Science*, vol. 4, no. 1, 2023.

[12] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, "Learning lyapunov functions for piecewise affine systems with neural network controllers," *arXiv preprint arXiv:2008.06546*, 2020.

[13] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[14] P. Grieder, "Efficient computation of feedback controllers for constrained systems," Ph.D. dissertation, ETH Zurich, 2004.

[15] T. Geyer, F. D. Torrisi, and M. Morari, "Optimal complexity reduction of polyhedral piecewise affine systems," *Automatica*, vol. 44, no. 7, pp. 1728–1740, 2008.

[16] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proceedings of the European Control Conference (ECC)*, 2013, pp. 502–510.

[17] J. Lofberg, "Yalmip : a toolbox for modeling and optimization in MATLAB," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 284–289.

[18] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

[19] L. Lüken, D. Brandner, and S. Lucia, "Sobolev training for data-efficient approximate nonlinear MPC," in *Proceedings of the 22th IFAC World Congress*, 2023, pp. 6290–6297.