Approximate Closed-Loop Robust Model Predictive Control With Guaranteed Stability and Constraint Satisfaction

Joel A. Paulson and Ali Mesbah, *Senior Member, IEEE*

***Abstract*—The real-time implementation of closed-loop robust model predictive control (MPC) schemes is an important challenge for fast systems, as their solution complexity depends strongly on the system size, control policy parametrization, and prediction horizon. We look to address this problem by approximating the implicitly-defined MPC controller using deep learning. Although the resulting neural network approximation has a small memory footprint and can be efficiently computed, it does not guarantee robust constraint satisfaction or stability. We propose a novel projection-based strategy that is capable of providing a certificate of robust feasibility and input-to-state stability in real-time. We also show how this projection operator can be formulated as a parametric quadratic program that is solvable offline. The advantages of the proposed approach are demonstrated on a benchmark case study.**

***Index Terms*—Robust model predictive control, deep neural networks, input-to-state stability, safe invariant sets.**

I. Introduction

Model predictive control (MPC) is a popular optimization-based strategy for control of constrained multivariable systems [1]. Emerging MPC applications, however, present unique challenges for the design of controllers with low enough computational and memory requirements for deployment in embedded systems [2]. These challenges can be significantly compounded when deploying robust/stochastic MPC formulations that explicitly account for some form of system uncertainty [3], [4].

There has been significant work on so-called “fast MPC”, which broadly includes development of: (i) fast solvers and tailored numerical implementations [5], and (ii) *explicit* MPC laws that are computed offline as a function of all feasible states [6]. Explicit MPC leverages the fact that, for linear time-invariant systems with quadratic costs, the MPC problem can be cast as a parametric quadratic program whose solution is a piecewise affine function defined on polytopes [6]. One important drawback of explicit MPC, however, is that the number of polytopic regions can grow exponentially with the number of constraints, which is further exacerbated in explicit robust MPC laws (e.g., [7], [8]).

Recently, there has been increasing interest in *approximate* MPC approaches, which aim to determine an explicit, cheap-to-evaluate representation of the controller using data generated from the offline solution of a MPC problem. Various function approximation schemes for deriving approximate MPC laws have been investigated including polynomials [9], radial basis functions [10], and deep neural networks [11]–[13]. These approaches generally result in good closed-loop performance; however, approximate MPC laws typically cannot *by design* guarantee closed-loop stability and feasibility. This remains a largely open challenge in approximate MPC, especially for uncertain systems.

This letter addresses the problem of guaranteeing closed-loop stability and feasibility of approximate robust MPC for linear systems subject to additive uncertainty. In particular, we look to approximate closed-loop robust MPC laws that seek to optimize over a class of control policies, as opposed to open-loop control inputs, in order to improve performance often at the expense of a significant increase in online computational cost. We use deep neural networks (DNNs) to approximate the expensive-to-evaluate closed-loop robust MPC law. This choice is motivated by several recent works that demonstrate particular advantages of DNNs in approximate MPC including their broad applicability as a consequence of the universal function approximation theorem [14], their ability to be stored and evaluated efficiently on low-cost embedded systems [12], and their ability to scale to problems with a large state dimension [15].

One promising approach for ensuring feasibility in DNN-based controllers is to project the output of the DNN into an appropriately defined invariant set, as recently discussed for deterministic linear systems in [11]. The main contribution of this letter is to propose a novel projection operator that can guarantee robust constraint satisfaction and input-to-state stability of DNN-approximated closed-loop robust MPC laws

Manuscript received December 10, 2019; revised February 9, 2020; accepted March 1, 2020. Date of publication March 12, 2020; date of current version May 25, 2020. Recommended by Senior Editor M. Guay. *(Corresponding author: Ali Mesbah.)*

Joel A. Paulson is with the Department of Chemical and Biomolecular Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: paulson.82@osu.edu).

Ali Mesbah is with the Department of Chemical and Biomolecular Engineering, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: mesbah@berkeley.edu).

Digital Object Identifier 10.1109/LCSYS.2020.2980479

in real-time. Furthermore, we demonstrate how this projection operator can be formulated as a parametric quadratic programming problem using variable lifting techniques such that an exact explicit solution can be obtained offline. The advantages of the proposed projected DNN control laws in terms of a substantially reduced online cost and memory footprint are demonstrated in a benchmark problem.

Notation: The sets of non-negative and positive integers and non-negative reals are denoted by $\mathbb{N}$, $\mathbb{N}_+$, and $\mathbb{R}_+$, respectively. Given $a, b \in \mathbb{N}$ such that $a < b$, we denote $\mathbb{N}_{[a,b]} := \{a, a+1, \dots, b\}$. We let $\|\cdot\|$ represent an arbitrary Hölder p-norm and $\mathcal{B}_r := \{x : \|x\| \le r\}$ denote a p-ball of radius $r \in \mathbb{R}_+$. Given a set X and a real matrix M of compatible dimensions (possibly a scalar), the image and the preimage of X under M are denoted by $MX := \{Mx : x \in X\}$ and $M^{-1}X := \{x : Mx \in X\}$, respectively. A set $X \subset \mathbb{R}^n$ is a *C-set* if it is compact, convex, and contains the origin in its interior. Given a C-set $X \subset \mathbb{R}^n$, the function $\Psi_X(x) := \inf\{\mu : x \in \mu X,\ \mu \ge 0\}$ is called the Minkowski function. A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ *belongs to class* $\mathscr{K}$ if it is continuous, strictly increasing, and $\varphi(0) = 0$. A function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ *belongs to class* $\mathscr{KL}$ if for each fixed $k \in \mathbb{R}_+$, $\beta(\cdot, k) \in \mathscr{K}$ and for each fixed $s \in \mathbb{R}_+$, $\beta(s, \cdot)$ is non-increasing and $\lim_{k\to\infty} \beta(s, k) = 0$.

II. Problem Statment

Consider a discrete-time, linear time-invariant system with an additive source of uncertainty

$$x^+ = Ax + Bu + w, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state at the current time instant, x^+ is the state at the next time instant, $u \in \mathbb{R}^m$ is the control input, and w is the bounded disturbance. The system is subject to hard constraints

$$(x, u, w) \in \mathcal{X} \times \mathcal{U} \times \mathcal{W}. \tag{2}$$

Assumption 1: The sets $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ are polytopic C-sets in $\mathbb{R}^n$, $\mathbb{R}^m$, and $\mathbb{R}^n$, respectively.

The objective of this letter is to design a controller that: (i) ensures the closed-loop system satisfies constraints for all times and for all possible disturbance sequences, (ii) guarantees robust stability of the origin, (iii) minimizes some specified cost function under uncertainty, and (iv) ensures the controller is implementable on resource-limited hardware for systems with fast sampling times. To achieve these objectives, we focus on closed-loop robust MPC methods that can mitigate conservative control performance by optimizing over a class of control policies. However, these robust MPC methods often lead to expensive-to-evaluate optimization problems that do not meet the fourth requirement above.

Let $\kappa_{\text{rmpc}} : \mathcal{X}_{\text{rmpc}} \to \mathcal{U}$ denote the "ideal" control law that is implicitly defined as the solution to some closed-loop robust MPC scheme with domain of attraction $\mathcal{X}_{\text{rmpc}} \subseteq \mathcal{X}$. It is assumed that a multiparametric solution for $\kappa_{\text{rmpc}}(\cdot)$ either cannot readily be obtained (e.g., due to nonlinear terms in the objective) or that it is too expensive to deploy on resource-limited hardware. Instead, we look to learn a more efficient representation of the control law using function approximation techniques. We focus on deep neural network (DNN) approximations of the form

$$\mathcal{N}(x; p) = \alpha_L \circ \beta_{L-1} \circ \alpha_{L-1} \circ \cdots \circ \beta_0 \circ \alpha_0(x), \tag{3}$$

where L is the number of hidden layers, $\alpha_0(x) = W_0 x + b_0$ is an affine transformation of the input, $\alpha_l(\xi_{l-1}) = W_l \xi_{l-1} + b_l$ (with $\xi_l \in \mathbb{R}^M$ and M denotes the number of nodes per hidden layer) are affine transformations of the hidden layers for $l \in \mathbb{N}_{[1,L]}$, β_l denotes the nonlinear activation functions (often chosen to be hyperbolic tangent functions or rectified linear units) for $l \in \mathbb{N}_{[0,L-1]}$, and $p = \{W_0, b_0, \dots, W_L, b_L\}$ are the collection of all unknown parameters in the network.

For a fixed network structure, the best approximation of κ_{rmpc} can be defined as the one that minimizes some loss function (e.g., mean squared error) of a given training data set. The training data can be generated offline by sampling the feasible states $x^i \in \mathcal{X}_{\text{rmpc}}$ and evaluating the control law $\kappa_{\text{rmpc}}(x^i)$ for all $i \in \mathbb{N}_{[1,N_s]}$. The approximated robust MPC control law is then given by

$$\kappa_{\text{dnn}}(x) = \mathcal{N}(x; p^\star) \approx \kappa_{\text{rmpc}}(x), \tag{4}$$

where $p^\star$ denotes the optimal network parameters. Unless the approximation error

$$\|\kappa_{\text{rmpc}}(x) - \kappa_{\text{dnn}}(x)\| \le \epsilon_{\text{approx}}, \quad \forall x \in \mathcal{X}_{\text{rmpc}}, \tag{5}$$

is systematically accounted for in the original robust MPC formulation, closed-loop guarantees may be lost whenever κ_{rmpc} is replaced by κ_{dnn}. This letter looks to address this problem for uncertain systems of the form (1). We propose to project the outputs of the DNN-based approximation to the closed-loop robust MPC law into an appropriately defined invariant set that also enforces a stability condition for the origin by design (i.e., for all possible ϵ_{approx}). The proposed projection, along with its relevant theoretical properties, are discussed in the next section. Since the proposed projection can be formulated as a relatively small quadratic program, we demonstrate how an explicit solution to this optimization can be derived offline in Section IV.

Remark 1: Although we focus on approximations of the form (3), the approach proposed in this letter is applicable to other approximate MPC methods.

III. Robust Feasibility and Stability of Deep Neural Network Control Laws

We can impose additional constraints on the approximated control law κ_{dnn} by projecting it into a suitably chosen set. For a given set $\mathcal{S} \subset \mathbb{R}^n$, we can formulate the projection operator as the solution to the optimization problem

$$\kappa(x, \mathcal{S}) = \operatorname{argmin}\{\|u - \kappa_{\text{dnn}}(x)\| : u \in \mathcal{S}\}. \tag{6}$$

This projection thus defines a family of control laws that are parametrized by the set $\mathcal{S}$, which can potentially vary with time. In this section, we look to analyze the properties of the nonlinear closed-loop system

$$x^+ = Ax + B\kappa(x, \mathcal{S}) + w, \tag{7}$$

under different choices of $\mathcal{S}$. In particular, we aim to select $\mathcal{S}$ such that certain robust feasibility and stability properties for system (7) can be guaranteed in real-time.

A. Real-Time Guaranteed Robust Constraint Satisfaction

We first recall a few standard definitions from set invariance theory [16].

Definition 1: A set Ω is *robust control invariant* (RCI) for system $x^+ = f(x, u, w)$ and constraint set $(\mathcal{X}, \mathcal{U}, \mathcal{W})$ if $\Omega \subseteq \mathcal{X}$ and if, for all $x \in \Omega$, there exists a $u \in \mathcal{U}$ such that $f(x, u, w) \in \Omega$ for all $w \in \mathcal{W}$.

Definition 2: A set Ω is *robust positively invariant* (RPI) for system $x^+ = f(x, w)$ and constraint set $(\mathcal{X}, \mathcal{W})$ if $\Omega \subseteq \mathcal{X}$ and if, for all $x \in \Omega$, $f(x, w) \in \Omega$ for all $w \in \mathcal{W}$.

Using these definitions, we can now establish the following important result.

Theorem 1: Let $\mathcal{C}$ be an RCI set for system (1) and constraint set (2). For each $x \in \mathcal{C}$, define the set-valued map

$$\mathcal{C}_u(x) = \{u \in \mathcal{U} : Ax + Bu + w \in \mathcal{C}, \forall w \in \mathcal{W}\}. \tag{8}$$

Then, for any initial condition $x \in \mathcal{C}$, the system (1) in closed-loop with the projected DNN-approximated controller $\kappa_{\text{fnn}}(x) := \kappa(x, \mathcal{C}_u(x))$ satisfies the constraints (2) for all times and all possible disturbances.

Proof: Since $\mathcal{C}$ is an RCI set, the optimization (6) must be recursively feasible for any $x \in \mathcal{C}$. By definition, any control law $\nu : \mathcal{C} \to \mathcal{U}$ that selects values from $\mathcal{C}_u(x)$, i.e., $\nu(x) \in \mathcal{C}_u(x)$, $\forall x \in \mathcal{C}$ guarantees that $\mathcal{C}$ is an RPI set for the system $x^+ = Ax + B\nu(x) + w$ and constraint set $(\mathcal{X}_\nu, \mathcal{U})$ where $\mathcal{X}_\nu = \mathcal{X} \cap \{x : \nu(x) \in \mathcal{U}\} \subseteq \mathcal{X}$. Since the projection (6) enforces this condition, the assertion must hold. ■

Generally, there are an infinite number of RCI sets for a given system. Since we want the control law (6) to have as large a region of attraction as possible, our attention is restricted to the *maximal* RCI set contained in $\mathcal{X}$, denoted by $\mathcal{C}_{\max}$. By definition, $\mathcal{X}_{\text{rmpc}} \subseteq \mathcal{C}_{\max}$, meaning the projection does not eliminate any feasible solutions of the original controller when $\mathcal{C}_{\max}$ is used in Theorem 1. This set can be computed as the limit of the recursion

$$\mathcal{C}_{i+1} = \text{Pre}(\mathcal{C}_i) \cap \mathcal{X}, \quad \mathcal{C}_0 = \mathcal{X}, \tag{9}$$

where $\text{Pre}(\Omega) = \{x \mid \exists u \in \mathcal{U} : Ax + Bu + w \in \Omega, \forall w \in \mathcal{W}\}$ denotes the predecessor set. If and only if $\mathcal{C}_{i^\star+1} = \mathcal{C}_{i^\star}$ for some $i^\star \in \mathbb{N}_+$, then $\mathcal{C}_{\max} = \mathcal{C}_{i^\star}$. Although (9) is not guaranteed to converge in finite-time, a variety of methods exist for finding invariant inner approximations of $\mathcal{C}_{\max}$. In addition, evaluation of the mapping $\text{Pre}(\Omega)$ can be done with readily available computational geometry packages such as MPT3 [17]. Even though these types of set operations have been traditionally limited to problems of relatively small dimension, recent work discusses ways that these approaches can be scaled to much larger dimensional problems using ideas from linear programming [18].

Theorem 1 addresses feasibility of the projection operator; however, feasibility does not imply stability. If the feasible set $\mathcal{C}$ is bounded, then one can think of the system (7) as being stable in a weak Lyapunov sense. However, one is often interested in obtaining stronger stability guarantees, which is discussed in more detail next.

B. Real-Time Guaranteed Input-to-State Stability (ISS)

Whenever the disturbance is non-zero, it is not possible to guarantee asymptotic stability of the origin. Instead, we take advantage of the notion of input-to-state stability (ISS) [19] that can be defined for system (7) as follows.

Definition 3: Let Ω be a subset of $\mathbb{R}^n$ containing the origin in its interior and let $\phi(k, x, w_{0:k-1})$ denote the solution to (7) at time k for a given state x at time 0 and disturbance sequence $w_{0:k-1} = \{w_0, \dots, w_{k-1}\}$. We call the system (7) *ISS in* Ω if there exists a $\mathscr{KL}$-function $\beta(\cdot, \cdot)$ and a $\mathscr{K}$-function $\gamma(\cdot)$ such that for all $x \in \Omega$ and all $\{w_j \in \mathcal{W}\}_{j \in \mathbb{N}_+}$, the corresponding state trajectories satisfy $\|\phi(k, x, w_{0:k-1})\| \leq \beta(\|x\|, k) + \gamma(\|w_{0:k-1}\|)$, $\forall k \in \mathbb{N}$.

We leverage the following result that establishes conditions under which ISS can be guaranteed.

Lemma 1 [19]: Let Ω be an RPI set for the system (7) and constraint set (2) that contains the origin in its interior. Furthermore, let there exist $\mathscr{K}$-functions $\alpha_1(\cdot)$, $\alpha_2(\cdot)$, $\alpha_3(\cdot)$, $\sigma(\cdot)$ and a continuous function $V : \mathbb{R}^n \to \mathbb{R}_+$ such that

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \tag{10a}$$

$$V(x^+) - V(x) \leq -\alpha_3(\|x\|) + \sigma(\|w\|), \tag{10b}$$

for all $x \in \Omega$, $w \in \mathcal{W}$, and $x^+ = Ax + B\kappa(x, \mathcal{S}) + w$. Then, the system (7) is ISS in Ω.

A function $V(\cdot)$ that satisfies the hypothesis of Lemma 1 is called an *ISS-Lyapunov function*.

To ensure the control law (6) induces some ISS-Lyapunov function for the closed-loop system (7), we rely on a particular type of invariant set, as defined next (see [16]).

Definition 4: Given a scalar $\lambda \in [0, 1)$, a C-set Ω is a λ-*contractive set* for the system $x^+ = Ax + Bu$ and constraint set $(\mathcal{X}, \mathcal{U})$ if $\Omega \subseteq \mathcal{X}$ and if, for all $x \in \Omega$, there exists a $u \in \mathcal{U}$ such that $Ax + Bu \in \lambda\Omega$.

Roughly speaking, a set $\mathcal{L}$ is λ-contractive if all states in $\mathcal{L}$ can be driven into a tighter region $\lambda\mathcal{L}$ by applying a one-step control input in the absence of disturbances. Note that a procedure similar to (9) can be used to derive the maximal λ-contractive set (see [20] for further details). Thus, this will be our vehicle for enforcing a contractive constraint in the proposed projection problem. We first provide some useful results concerning the Minkowski functions of C-sets.

Lemma 2 [21]: Let X and Y be C-sets in $\mathbb{R}^n$ such that $Y \subseteq X$. Then, (i) $\Psi_X(x) \leq \Psi_Y(x)$, (ii) $\Psi_{\mathcal{B}_r}(x) = r^{-1}\|x\|$, and (iii) $\Psi_X(x + y) \leq \Psi_X(x) + \Psi_X(y)$ for all $x \in \mathbb{R}^n$.

Theorem 2: Let $\mathcal{L}$ be a λ-contractive set for the system (1) with $w = 0$ and the constraint set $(\mathcal{X}, \mathcal{U})$. For each $x \in \mathcal{L}$, define the set-valued map

$$\mathcal{L}_u(x) = \{u \in \mathcal{U} : Ax + Bu \in \lambda\Psi_{\mathcal{L}}(x)\mathcal{L}\}. \tag{11}$$

Furthermore, let $\mathbb{X}_f \subseteq \mathcal{L}$ be an RPI set for the closed-loop system (7) and constraint set (2) that contains the origin in its interior. Then, the closed-loop system $x^+ = Ax + B\kappa_{\text{snn}}(x) + w$ is ISS in $\mathbb{X}_f$, where $\kappa_{\text{snn}}(x) := \kappa(x, \mathcal{L}_u(x))$.

Proof: The proof involves showing that the Minkowski function $V(x) = \Psi_{\mathcal{L}}(x)$ is an ISS-Lyapunov function for $x^+ = Ax + B\kappa_{\text{snn}}(x) + w$. Since $\mathcal{L}$ is non-empty, there exist constants $0 < r_2 < r_1$ such that $\mathcal{B}_{r_2} \subset \mathcal{L} \subset \mathcal{B}_{r_1}$. From Lemma 2(i, ii), this implies $c_1\|x\| \leq \Psi_{\mathcal{L}}(x) \leq c_2\|x\|$, with

$c_1 = r_1^{-1}$ and $c_2 = r_2^{-1}$ such that $V(\cdot)$ satisfies (10a). Next, the constraints (11) enforce the following for all $x \in \mathbb{X}_f$

$$\Psi_{\mathcal{L}}(Ax + B\kappa_{\text{snn}}(x)) \leq \lambda \Psi_{\mathcal{L}}(x).$$

Moreover, the inequality $\Psi_{\mathcal{L}}(x^+) - \Psi_{\mathcal{L}}(Ax + B\kappa_{\text{snn}}(x)) \leq \Psi_{\mathcal{L}}(w)$ must hold based on Lemma 2(iii). By combining these two inequalities and (10a), it follows that

$$\Psi_{\mathcal{L}}(x^+) - \Psi_{\mathcal{L}}(x) \leq -\alpha_3(\|x\|) + \sigma(\|w\|),$$

where $\alpha_3(\|x\|) = (1-\lambda)c_1\|x\|$ and $\sigma(\|w\|) = c_2\|w\|$. The statement then follows from Lemma 1. ■

Theorem 2 assumes robust feasibility of a subset of the state space, which cannot be guaranteed a priori in general. One way to recover this property is to select $\mathcal{S} = \mathcal{C}_u(x) \cap \mathcal{L}_u(x)$; however, these constraints are not guaranteed to be recursively feasible. In practice, the stability constraint $u \in \mathcal{L}_u(x)$ can be softened using the exact penalty function method [22], which ensures that robust feasibility holds (Theorem 1) and the closed-loop system is ISS in any RPI subset of $\mathcal{C} \cap \mathcal{L}$ (Theorem 2).

The stability result in Theorem 2 can be strengthened in the absence of disturbances, i.e., $w = 0$ in system (1).

Corollary 1: For any initial condition $x_0 \in \mathcal{L}$, where the set $\mathcal{L}$ satisfies the conditions of Theorem 2, the origin of the closed-loop system $x_{k+1} = Ax_k + B\kappa_{\text{snn}}(x_k)$ is asymptotically stable and satisfies constraints $(x_k, \kappa_{\text{snn}}(x_k)) \in \mathcal{X} \times \mathcal{U}$ for all times $k \in \mathbb{N}$.

The proof of this result is omitted due to space limitations but follows from the ISS property with $\mathcal{W} = \{0\}$ and recursive feasibility of the control law $\kappa_{\text{snn}}(x)$ according to established properties of λ-contractive sets [16].

IV. Offline Multiparametric Solution to the Projection Optimization

The projection operator (6) is implicitly defined as the solution to an optimization problem whose structure depends on the chosen set $\mathcal{S}$. Under Assumption 1, the control input, the maximal RCI, and the maximal λ-contractive sets are all polytopes that can be written in terms of their irreducible halfspace representations. This implies that the constraints $\mathcal{S}$ under both Theorems 1 and 2 are linear such that (6) is a quadratic program (QP) when $\|\cdot\|_2^2$ is used in the objective.

The current state x is a parameter in the projected control law (6), similar to MPC. This problem can be solved *online* using one of the many QP solvers tailored to embedded applications, e.g., [23], [24], or *offline* using multiparametric quadratic programming (mpQP) algorithms, e.g., [6]. The latter approach can drastically reduce the memory and computational load when the dimensions of the QP are fairly small [25]. However, in such a case when an explicit solution is desired, we are unable to directly apply mpQP algorithms to the projection since x does not appear linearly in (6). We can address this challenge by defining an equivalent problem in a lifted space that depends on $\mathcal{S}$. For $\mathcal{S} = \mathcal{C}_u(x)$, the nonlinearity $\tilde{u} = \kappa_{\text{dnn}}(x)$ is fixed given the current state, so we can compute an explicit solution with respect to the "lifted" parameter vector $\theta = [x^\top, \tilde{u}^\top]^\top$. Accordingly, the projection can be expressed in standard form

$$\begin{aligned} \min_{u} \quad & \frac{1}{2} u^\top H u + \theta^\top F u, \\ \text{s.t.} \quad & Gu \leq b + S\theta, \quad \theta \in \Theta, \end{aligned} \tag{12}$$

where $\Theta = \mathcal{C} \times \mathcal{U}_{\text{dnn}}$ is the set of parameters, $\mathcal{U}_{\text{dnn}} \supseteq \kappa_{\text{dnn}}(\mathcal{C})$ is a bounding box for the DNN outputs, and H, F, G, S, b can be derived from standard algebraic manipulations. The solution $u^\star : \Theta \to \mathcal{U}$ to this mpQP is a piecewise affine (PWA) function of the form [6]

$$u^\star(\theta) = K_i\theta + h_i \quad \text{if } E_i\theta \leq e_i, \tag{13}$$

where the polyhedral sets $\{\theta : E_i\theta \leq e_i\}_{i=1}^R$ are a partition of Θ composed of R critical regions. Thus, the proposed robustly feasible control law $\kappa_{\text{fnn}}(x)$ in Theorem 1 is the convolution of a DNN with a PWA function that can be obtained by substituting $\theta \leftarrow [x^\top, \kappa_{\text{dnn}}(x)^\top]^\top$ into (13). Note that a similar result holds for the stabilizing control law $\kappa_{\text{snn}}(x)$ in Theorem 2 when the parameter vector is chosen as $\theta = [x^\top, \tilde{u}^\top, \alpha]^\top$, with $\alpha = \Psi_{\mathcal{L}}(x)$. Given the polytopic set $\mathcal{L} = \{x : F_L x \leq \mathbf{1}\}$, the Minkowski function can be evaluated as $\Psi_{\mathcal{L}}(x) = \max(F_L x)$ where $\mathbf{1}$ is a vector of all ones and $\max(\cdot)$ is the maximum element of any vector.

The number of critical regions defining the PWA solution of an mpQP may increase exponentially with the number of constraints, leading to heavy memory and computational loads. This is known to be an important limitation of explicit MPC [6], especially for problems with long prediction horizons. The proposed controllers, however, overcome this challenge since the number of constraints in the projection (6) is independent of the prediction horizon. Furthermore, if needed, the number of constraints defining the invariant sets can be reduced [26] at the cost of a smaller feasible region. We also note that the proposed approach is applicable to MPC problems for which explicit solutions cannot readily be obtained such as those with nonlinear objectives and/or nonlinear parametrized feedback policies in the prediction.

Remark 2: In the absence of state constraints, $\mathcal{L}_u(x) = \mathcal{U}$. When $\mathcal{U}$ is composed of box constraints, then the projection in Theorem 1 reduces to a simple saturation operation.

Remark 3: The idea of explicitly solving the Euclidean projection onto non-trivial polyhedral constraints with mpQP methods has been explored in [27], which shows that the resulting PWA function has particular structure that can be exploited for efficient online evaluation. However, it is unknown if (and when) this explicit solution is preferred over state-of-the-art embedded QP solvers unless the memory and worst-case complexity are exactly certified. This certification problem, along with new approaches that combine explicit and implicit ideas, has been recently addressed in [25], [28] for particular QP solution methods.

V. Case Study

The following double integrator is a benchmark problem in the robust MPC literature

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u + w, \tag{14}$$

where $\mathcal{W} = \{w \in \mathbb{R}^2 : \|w\|_\infty \leq 1.5\}$, $\mathcal{U} = \{u \in \mathbb{R} : \|u\|_\infty \leq 5\}$, and $\mathcal{X} = \{x \in \mathbb{R}^2 : x \in [-50, 10] \times [-50, 10]\}$ correspond to the constraint sets in (2). We consider the ideal control law κ_{rmpc} to be the solution to a multi-stage MPC problem, which optimizes over a feedback policy defined in terms of a scenario tree [29]. This implies that the disturbance set must be approximated by a discrete number of scenarios $\mathcal{W} \approx \tilde{\mathcal{W}} = \{w^1, \ldots, w^s\}$, where $s \in \mathbb{N}$ denotes the total number of scenarios. Here, we take all combinations of the minimum, nominal, and maximum values of the disturbance such that $s = 3^2 = 9$. The optimization problem solved at each sampling time is then given by

$$\begin{aligned}\min_{U} \; & \sum_{k=0}^{N-1}\sum_{i=1}^{s^k} \ell(x_k^i, u_k^i) + \sum_{i=1}^{s^N} \ell_f(x_N^i),\\ \text{s.t.} \; & \tilde{X}_{k+1} = \bigcup_{i=1}^{s^k}\bigcup_{j=1}^{s} Ax_k^i + Bu_k^i + w^j,\\ & \tilde{X}_0 = \{x\}, \; (\tilde{X}_k, \tilde{U}_k) \subseteq \mathcal{X} \times \mathcal{U}, \; \tilde{X}_N \subseteq \mathcal{X}_f,\\ & \forall k = 0, \ldots, N-1, \end{aligned} \tag{15}$$

where $N \in \mathbb{N}$ is the prediction horizon, $\tilde{X}_k = \{x_k^i\}_{i=1}^{s^k}$ is a point-wise approximation of the reachable state set at time k, $\tilde{U}_k = \{u_k^i\}_{i=1}^{s^k}$ is the set of inputs in the scenario tree at time k, $U = \{\tilde{U}_0, \ldots, \tilde{U}_{N-1}\}$, $\ell(\cdot)$ is the stage cost function, $\ell_f(\cdot)$ is the terminal cost function, and $\mathcal{X}_f$ is the terminal set. To avoid exponential growth in the scenario tree size with respect to N, we consider branching in the tree only up until a certain stage, often referred to as the robust horizon N_r. Thus, we let $\kappa_{\text{rmpc}}(x) = u_0^{1\star}(x)$ denote the ideal multi-stage control law with design parameters: $N = 10$, $N_r = 2$, $\ell(x, u) = \|x\|_2^2 + 0.01\|u\|_2^2$, $\ell_f(x) = 0$, and $\mathcal{X}_f = \mathcal{X}$. Let $\mathbb{X}_F$ denote the feasible region of (15). These chosen design parameters do not guarantee recursive feasibility in $\mathbb{X}_F$; however, it was verified through simulations that $\mathcal{C}_{\max} \subseteq \mathbb{X}_F$ is RPI for the closed-loop system $x^+ = Ax + B\kappa_{\text{rmpc}}(x) + w$ and constraints (2) such that $\mathcal{X}_{\text{rmpc}} = \mathcal{C}_{\max}$ is the relevant region of attraction, which can be calculated using (9).

We look to obtain DNN approximations of the form (4) to the ideal multistage control law. Motivated by [12], the activation functions were chosen to be rectified linear units. Training data was generated by solving (15) at $N_s = 1000$ points randomly sampled in $\mathcal{X}_{\text{rmpc}}$ using the interior point solver IPOPT [30]. The DNN was trained using the Levenberg-Marquardt algorithm in the Deep Learning Toolbox [31], and κ_{fnn} was constructed using $\mathcal{C} = \mathcal{C}_{\max}$ in Theorem 1. The resulting projection operator was formulated as a mpQP (12) and solved with MPT3 [17], which generated a PWA function with $R = 11$ regions. All computations were performed in MATLAB R2019a on a MacBook Pro with 32 GB of RAM and 2.3 GHz Intel i9 processor.

Since the network structure has a significant influence on training, results were compiled for various numbers of nodes M and layers L in Table I, where $ML = 30$ to approximately fix the network complexity. We define the mean squared error (MSE) between κ_{rmpc} and an approximate control law $\hat{\kappa}$ as

$$\text{MSE} = \frac{1}{N_{\text{val}}}\sum_{i=1}^{N_{\text{val}}} \|\kappa_{\text{rmpc}}(x^i) - \hat{\kappa}(x^i)\|_2^2, \tag{16}$$

where $\{x^i\}_{i=1}^{N_{\text{val}}}$ is a set of validation points randomly sampled from $\mathcal{X}_{\text{rmpc}}$ and $N_{\text{val}} = 1000$ is the total number of validation points. The MSE provides a measure of accuracy for $\hat{\kappa}(x)$ averaged across the feasible region of the controller. From Table I, we can see that $(M, L) = (6, 5)$ yields the lowest MSE, suggesting that one should strike a balance between network width and depth. To demonstrate the key advantage of the proposed projection, we also report the percentage of feasible closed-loop trajectories under the approximate control law. We define this quantity as follows. Let $\{w_k^i \in \mathcal{W}\}_{i=1}^{N_{\text{val}}}$ be a set of randomly sampled disturbance values for all $k \in \mathbb{N}_{[0,N_{\text{sim}}]}$ where $N_{\text{sim}} = 50$ is the number of simulation time steps. The closed-loop system under the approximate control law then evolves as

$$x_{k+1}^i = Ax_k^i + B\hat{\kappa}(x_k^i) + w_k^i, \; x_0^i = [-50, -3.5]^\top, \tag{17}$$

where the initial state is a vertex of $\mathcal{C}_{\max}$ that leads to active state constraints and thus a reasonably high probability of constraint violation. The i^{th} trajectory is feasible if it satisfies $(x_k^i, \hat{\kappa}(x_k^i)) \in \mathcal{X} \times \mathcal{U}$ for all $k \in \mathbb{N}_{[0,N_{\text{sim}}]}$. From Table I, we see that κ_{dnn} results in closed-loop state constraint violation in all considered cases, whereas κ_{fnn} provides a robust feasibility certificate by design. We emphasize that this is an essential property to guarantee in safety-critical applications and that it holds regardless of the DNN approximation error.

The online evaluation times of the approximate MPC laws, averaged over the set of random validation samples, are also shown Table I. For comparison purposes, the ideal control law κ_{rmpc} took 314.6 ± 78.4 ms to be evaluated using IPOPT versus 0.76 ± 0.09 ms for κ_{fnn} with $(M, L) = (6, 5)$, representing a speedup of over 400 times. Fig. 1 shows that the

TABLE I
APPROXIMATE CLOSED-LOOP ROBUST MPC PERFORMANCE METRICS FOR VARIOUS NETWORK STRUCTURES

(M, L)	mse $\times 10^4$	feas. [%]	cpu time [ms]	mem. [kB]
	dnn | fnn	dnn | fnn	dnn | fnn	dnn | fnn
(30,1)	386 | 375	91 | 100	0.30 | 0.82	0.94 | 2.25
(15,2)	8.3 | 7.5	96 | 100	0.28 | 0.76	2.35 | 3.36
(6,5)	4.4 | 4.4	97 | 100	0.26 | 0.76	1.51 | 2.82
(2,15)	8548 | 3173	0 | 100	0.28 | 0.78	0.73 | 2.04

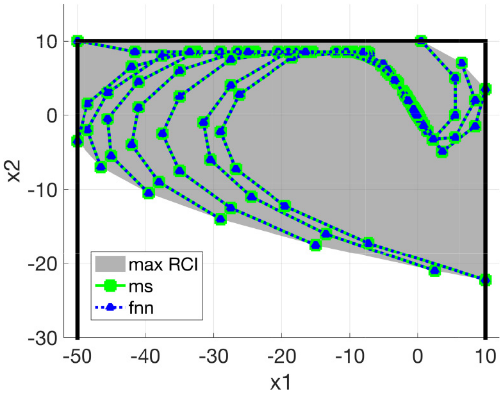

Fig. 1. Phase plot of the closed-loop state profiles for the double integrator under the ideal multistage control law (ms) and the proposed approximate MPC law from Theorem 1 with $(M, L) = (6, 5)$ (fnn). The disturbance is zero and results are shown for initial states at the vertices of $\mathcal{C}_{\max}$.

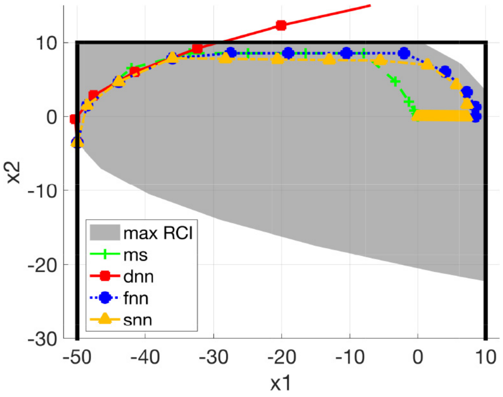

Fig. 2. Phase plot of the closed-loop profiles for the double integrator under the ideal multistage control law (ms), the poorly trained DNN-approximated MPC law with $(M, L) = (2, 15)$ (dnn), and the proposed approximate MPC laws from Theorem 2 (fnn) and Theorem 2 (snn).

closed-loop behavior is nearly identical for these two control laws whenever the system is initialized at each vertex of $\mathcal{C}_{\max}$, which suggests that the improvement in online cost and memory footprint did not come at the cost of a loss in performance. In addition, as shown in Fig. 2, the closed-loop system is unstable for κ_{dnn} with $(M, L) = (2, 15)$, which is a direct consequence of the large MSE. Although κ_{fnn} is able to ensure constraints are satisfied, it is unable to drive the system to the origin, even in the case of zero disturbance. By implementing κ_{snn} from Theorem 2 ($\lambda = 0.99$), we are able to guarantee the closed-loop system is ISS, with only minor increases in online cost and memory footprint.

VI. Conclusion and Future Work

This letter addresses the problem of approximating computationally expensive closed-loop robust MPC laws using deep neural networks. A real-time, projection-based strategy is developed for ensuring robust feasibility and input-to-state stability of the closed-loop system under the approximated control law, which is essential for safety-critical applications with fast sampling times. We also show that multiparametric quadratic programming algorithms can be used to solve the projection problem fully offline. Future work will focus on ways to extend this projection to larger-scale and nonlinear systems as well as its efficient implementation on resource-limited embedded control hardware.

References

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

[2] T. A. Johansen, “Toward dependable embedded model predictive control,” *IEEE Syst. J.*, vol. 11, no. 2, pp. 1208–1219, Jun. 2017.

[3] A. Bemporad and M. Morari, “Robust model predictive control: A survey,” in *Robustness in Identification and Control*. London, U.K.: Springer, 1999, pp. 207–226.

[4] A. Mesbah, “Stochastic model predictive control: An overview and perspectives for future research,” *IEEE Control Syst. Mag.*, vol. 36, no. 6, pp. 30–44, Dec. 2016.

[5] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.

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

[7] M. de la Pena, A. Bemporad, and C. Filippi, “Robust explicit MPC based on approximate multi-parametric convex programming,” in *Proc. 43rd IEEE Conf. Decis. Control*, Nassau, Bahamas, 2004, pp. 2491–2496.

[8] E. N. Pistikopoulos, N. P. Faísca, K. I. Kouramas, and C. Panos, “Explicit robust model predictive control,” *IFAC Proc. Vol.*, vol. 42, no. 11, pp. 243–248, 2009.

[9] A. Chakrabarty, V. Dinh, M. J. Corless, A. E. Rundell, S. H. Żak, and G. T. Buzzard, “Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences,” *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 135–148, Jan. 2017.

[10] L. H. Csekő, M. Kvasnica, and B. Lantos, “Explicit MPC-based RBF neural network controller design with discrete-time actual Kalman filter for semiactive suspension,” *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1736–1753, Sep. 2015.

[11] S. Chen *et al.*, “Approximating explicit model predictive control using constrained neural networks,” in *Proc. Amer. Control Conf.*, Milwaukee, WI, USA, 2018, pp. 1520–1527.

[12] B. Karg and S. Lucia, “Efficient representation and approximation of model predictive control laws via deep learning,” 2018, [Online]. Available: arXiv:1806.10644.

[13] A. D. Bonzanini, J. A. Paulson, D. B. Graves, and A. Mesbah, “Safe dose delivery in fast sampling atmospheric plasmas using projection-based approximate economic MPC,” in *Proc. IFAC World Congr.*, Berlin, Germany, 2020.

[14] A. R. Barron, “Universal approximation bounds for superpositions of a sigmoidal function,” *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, May 1993.

[15] S. W. Chen, T. Wang, N. Atanasov, V. Kumar, and M. Morari, “Large scale model predictive control with neural networks and primal active sets,” 2019, [Online]. Available: arXiv:1910.10835.

[16] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Cham, Switzerland: Springer, 2008.

[17] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, “Multi-parametric toolbox 3.0,” in *Proc. Eur. Control Conf.*, Zürich, Switzerland, 2013, pp. 502–510.

[18] M. Fiacchini and M. Alamir, “Computing control invariant sets in high dimension is easy,” 2018, [Online]. Available: arXiv:1810.10372.

[19] L. Magni, D. M. Raimondo, and R. Scattolini, “Regional input-to-state stability for nonlinear model predictive control,” *IEEE Trans. Autom. Control*, vol. 51, no. 9, pp. 1548–1553, Sep. 2006.

[20] M. S. Darup and M. Cannon, “On the computation of λ-contractive sets for linear constrained systems,” *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1498–1504, Mar. 2017.

[21] S. V. Raković and M. Lazar, “Minkowski terminal cost functions for MPC,” *Automatica*, vol. 48, pp. 2721–2725, Oct. 2012.

[22] E. C. Kerrigan, “Robust constraint satisfaction: Invariant sets and predictive control,” Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 2001.

[23] S. Richter, C. N. Jones, and M. Morari, “Computational complexity certification for real-time MPC with input constraints based on the fast gradient method,” *IEEE Trans. Autom. Control*, vol. 57, no. 6, pp. 1391–1403, Jun. 2012.

[24] H. J. Ferreau, H. G. Bock, and M. Diehl, “An online active set strategy to overcome the limitations of explicit MPC,” *Int. J. Robust Nonlinear Control*, vol. 18, pp. 816–830, May 2008.

[25] G. Cimini and A. Bemporad, “Exact complexity certification of active-set methods for quadratic programming,” *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6094–6109, Dec. 2017.

[26] T. B. Blanco, M. Cannon, and B. De Moor, “On efficient computation of low-complexity controlled invariant sets for uncertain linear systems,” *Int. J. Control*, vol. 83, no. 7, pp. 1339–1346, 2010.

[27] S. Merkli, J. Jerez, A. Domahidi, R. S. Smith, and M. Morari, “Circuit generation for efficient projection onto polyhedral sets in first-order methods,” in *Proc. Eur. Control Conf.*, Linz, Austria, 2015, pp. 3440–3445.

[28] G. Cimini and A. Bemporad, “Complexity and convergence certification of a block principal pivoting method for box-constrained quadratic programs,” *Automatica*, vol. 100, pp. 29–37, Feb. 2019.

[29] S. Lucia, T. Finkler, and S. Engell, “Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty,” *J. Process Control*, vol. 23, no. 9, pp. 1306–1319, 2013.

[30] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Math. Program.*, vol. 106, pp. 25–57, 2006.

[31] *MATLAB and Deep Learning Toolbox Release 2019a*, MathWorks Inc., Natick, MA, USA, 2019.