An Adaptive Correction Scheme for Offset-Free Asymptotic Performance in Deep Learning-based Economic MPC ⋆

Dinesh Krishnamoorthy * **Ali Mesbah** ** **Joel A. Paulson** ***

* *Department of Chemical Engineering, Norwegian University of Science & Technology (NTNU), NO-7491 Trondheim, Norway (e-mail: dinesh.krishnamoorthy@ntnu.no).*
** *Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA 94720 USA (e-mail: mesbah@berkeley.edu)*
*** *Department of Chemical and Biomolecular Engineering, Ohio State University, Columbus, OH 43210 USA (e-mail: paulson.82@osu.edu).*

Abstract: There has been an increasing interest in explicit and cheap-to-evaluate control policies that approximate (computationally expensive) control laws such as model predictive control (MPC). However, approximate control policies are subject to approximation errors, leading to asymptotic performance losses. The contribution of this paper is three-fold: (i) a closed-loop training scheme is presented for deep neural network approximation of economic MPC; (ii) an online adaptive correction scheme is presented to account for the performance losses induced by approximation errors; and (iii) an offline performance verification scheme is presented to ensure that the approximate control policy converges to an equilibrium point of the system. The proposed approach is illustrated using a Williams-Otto reactor problem.

Keywords: Approximate MPC, deep neural networks, probabilistic guarantees, adaptive correction, offset-free performance

1. INTRODUCTION

Economic model predictive control (MPC) is a popular optimization-based control strategy that has found use in various applications. However, some of the open challenges with economic MPC include high memory footprint and high computational burden associated with repeatedly solving an optimal control problem online. Explicit MPC was proposed to address these challenges for the tracking problem, where the online optimal control law is replaced with a pre-computed control law (Bemporad et al., 2002; Tøndel et al., 2003). However, this approach can quickly become computationally intractable for large systems, and the extension to nonlinear systems with economic objectives is not trivial.

Recently, there has been an increasing interest to "approximate" MPC policies using suitable function approximators such as polynomials (Chakrabarty et al., 2017), radial-basis functions (Csekő et al., 2015), and deep neural networks (Karg and Lucia, 2020; Chen et al., 2018; Paulson and Mesbah, 2020a; Hertneck et al., 2018; Drgoňa et al., 2018; Zhang et al., 2019; Hirose et al., 2018). The underlying idea of approximate MPC is as follows. The optimal control problem is solved offline for various state realizations x_i to obtain the MPC policy $u_i^* = \pi_{mpc}(x_i)$. Then, $\pi_{mpc}(\cdot)$ is approximated using a training dataset consisting of N_s samples $\{(x_i, u_i^*)\}_{i=1}^{N_s}$ to derive a cheap-to-evaluate control policy $\pi_{approx}(\cdot)$. This can be seen as a policy approximation using an expert-based supervised learning (Bertsekas, 2019, Section 5.7), where $\pi_{approx}(\cdot)$ is the approximate policy function.

Although approximate MPC has been shown to give good closed-loop performance in many examples, there are inevitable approximation errors due to (i) the choice of the hyperparameters that decide the functional form and architecture of the approximate control policy, and (ii) possibly insufficient training data in some regions of the feasible state space leading to poor generalization. In addition, changes in some of the model parameters, or operation in parts of the state-space not covered adequately in the training dataset may affect the accuracy of the approximate control policy. These approximation errors can lead to asymptotic losses in closed-loop performance.

In this paper, we propose a closed-loop training procedure for approximating economic MPC using deep neural networks. We then propose a KKT-derived online adaptive correction scheme to account for the performance losses due to the policy approximation. To ensure convergence of the online adaptive correction scheme, which is not guaranteed by design, we propose an offline performance verification method. The performance verification provides probabilistic guarantees that the closed-loop control system converges to a stable equilibrium point.

The reminder of the paper is organized as follows. Section 2 presents the economic MPC problem, along with its

⋆ DK gratefully acknowledges the financial support from the IKTPLUSS project from the Research Council of Norway, and Peder Sather Grant. DK and JP contributed equally to this work.

asymptotic properties. Section 3 presents the approximate MPC scheme with closed-loop training. The proposed KKT-derived online adpative correction scheme for offset-free asymptotic performance and the probabilistic verification are presented in Section 4.

2. PRELIMINARIES: ECONOMIC MPC WITH STABILITY GUARANTEES

Consider a discrete-time nonlinear system

$$x(t+1) = f(x(t), u(t)), \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ denotes the states, $u \in \mathbb{R}^{n_u}$ denotes the control input, and the mapping $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ represents the system model. We look to obtain a feedback control law $\pi_{mpc} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ that is implicitly defined by solving an economic MPC problem. The corresponding optimal control problem (OCP) is formulated as

$$\begin{aligned}
V(x_0) = \min_{x(\cdot|t),u(\cdot|t)} \ & \sum_{k=0}^{N-1} \ell(x(k|t), u(k|t)) && \text{(2a)}\\
\text{s.t. } & x(k+1|t) = f(x(k|t), u(k|t)), \ \forall k \in \mathbb{I}_{0:N-1} && \text{(2b)}\\
& x(k|t) \in \mathcal{X}, \quad u(k|t) \in \mathcal{U}, \qquad \forall k \in \mathbb{I}_{0:N-1} && \text{(2c)}\\
& x(N|t) = x_e && \text{(2d)}\\
& x(0|t) = x_0, && \text{(2e)}
\end{aligned}$$

where $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ denotes the stage cost, which may include a combination of tracking and economic terms; N is the length of the prediction horizon; (2c) are the path constraints; (2d) is the terminal constraint; and (2e) is the current state measurement $x_0 = x(t)$ that is used to initialize the model. Note that (2d) is in the form of a terminal equality constraint, in which $x_e \in \mathcal{X}$ denotes a feasible equilibrium point (see Definition 2).

The optimal control problem (2) is repeatedly solved at each sample time t using $x(t)$ as the initial state, and the first sample of the optimal input $u^*(0|t)$ is implemented in a receding-horizon fashion. This implicitly leads to the following definition of the control law

$$u^*(t) = \pi_{mpc}(x(t)). \tag{3}$$

Definition 1. (Admissible set) The admissible set is defined as the set of pairs $(x_0, u(\cdot|t))$

$$\mathcal{Z}_N := \{(x_0, u(\cdot|t)) \mid \exists x(1|t), \ldots, x(N|t) : \text{(2b)–(2d)}\},$$

and the set of admissible states $\mathcal{X}_N = \text{Proj}_{\mathcal{X}}(\mathcal{Z}_N)$ can be defined as the projection of $\mathcal{Z}_N$ onto $\mathcal{X}$.

Definition 2. (Equilibrium point) A point (x_e, u_e) is called an equilibrium point if $x_e = f(x_e, u_e)$ holds. Further, (x_e, u_e) is called an optimal equilibrium point if

$$\ell(x_e, u_e) \leq \ell(x, u) \ \forall (x, u) \in \mathcal{X} \times \mathcal{U} \text{ with } x = f(x, u).$$

Definition 3. (Dissipativity) The system (1) is said to be dissipative w.r.t. an equilibrium point $(x_e, u_e) \in \mathcal{X} \times \mathcal{U}$ if there exists a storage function $\lambda : \mathcal{X} \to \mathbb{R}^+$ such that

$$\lambda(f(x,u)) - \lambda(x) \leq \ell(x,u) - \ell(x_e, u_e), \ \forall (x,u) \in \mathcal{X} \times \mathcal{U}. \tag{4}$$

Furthermore, system (1) is said to be strictly dissipative, if there exists $\alpha \in \mathcal{K}_\infty$ such that

$$\lambda(f(x,u)) - \lambda(x) \leq \ell(x,u) - \ell(x_e, u_e) - \alpha(\|x - x_e\|). \tag{5}$$

Assumption 1. (a) $\mathcal{X} \times \mathcal{U}$ is compact, and the functions $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ and $\ell : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ are continuous.

(b) Weak controllability - There exists a function $\gamma(\cdot)$ of class $\mathcal{K}_\infty$ such that for each $x \in \mathcal{X}_N$ there exists a feasible $u(\cdot|t)$ with $\sum_{k=0}^{N-1} |u(k|t) - u_e| \leq \gamma(\|x - x_e\|)$.

Assumption 2. Linear inequality constraint qualification and strong second order sufficient conditions holds for the optimal control problem (2).

Assumption 3. (a) There exists at least one equilibrium pair $(x_e, u_e) \in \mathcal{X} \times \mathcal{U}$.

(b) There exists a bounded non-negative storage function λ and $\alpha \in \mathcal{K}_\infty$ such that the OCP (2) is strictly dissipative w.r.t. the equilibrium point (x_e, u_e) in the sense of Definition 3.

Lemma 1. Given Assumptions 2 and 3, (x_e, u_e) is a unique minimizer of the steady-state optimization problem

$$\begin{aligned}
(x_e, u_e) = \underset{x,u}{\text{argmin}} \ & \ell(x, u) && \text{(6a)}\\
\text{s.t. } & x = f(x, u) && \text{(6b)}\\
& x \in \mathcal{X}, u \in \mathcal{U}. && \text{(6c)}
\end{aligned}$$

Proof Suppose another equilibrium point (x'_e, u'_e) exists that minimizes (6), this would imply $\ell(x'_e, u'_e) < \ell(x_e, u_e)$. However, evaluating the dissipation inequality (5) at (x'_e, u'_e) leads to a contradiction. Hence $\ell(x'_e, u'_e) < \ell(x_e, u_e)$ cannot be true, and $\ell(x_e, u_e)$ is the unique global minimizer of (6). □

Strict dissipativity implies that the system is optimally operated at steady-state. That is, the inequality

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{k=0}^{T-1} \ell(x_\pi(k; x_0), \pi_{mpc}(x_\pi(k; x_0))) \geq \ell(x_e, u_e)$$

holds for all $x \in \mathcal{X}$ and all admissible control sequences $\pi_{mpc}(x)$, as shown in (Grüne and Pannek, 2017, Proposition 8.9) and (Angeli et al., 2011, Proposition 6.4). A direct implication of this inequality is that, for a dissipative system, the closed-loop performance must be infinite horizon averaged optimal. That is,

$$\ell(x_\pi(k; x_0), \pi_{mpc}(x_\pi(k; x_0))) \to \ell(x_e, u_e) \text{ as } k \to \infty. \tag{7}$$

As shown by Diehl et al. (2010), one can use a rotated stage cost and the Lyapunov stability framework to show that (x_e, u_e) is indeed an asymptotically stable point of the closed-loop system using the economic MPC policy $\pi_{mpc}(\cdot)$. Defining the rotated stage cost as

$$L(x, u) := \ell(x, u) + \lambda(x) - \lambda(f(x, u)), \tag{8}$$

we can write the OCP as

$$\tilde{V}(x_0) = \left\{ \min_{x(\cdot|t),u(\cdot|t)} \sum_{k=0}^{N-1} L(x(k|t), u(k|t)) \mid \text{(2b)} - \text{(2e)} \right\}$$

and, as shown by Angeli et al. (2011), rotation does not alter the optimal solution. Hence, we can use the Lyapunov framework using the rotated stage cost.

Theorem 1. (Nominal stability of economic MPC policy). Let Assumptions 1, 2 and 3 hold. Then, the closed-loop system $x(t+1) = f(x(t), \pi_{mpc}(x(t)))$ is asymptotically stable at (x_e, u_e) with a region of attraction $\mathcal{X}_N$.

Proof See Appendix A. □

3. DEEP LEARNING-BASED APPROXIMATE MPC

To replace the implicitly-defined MPC control law (3) with a more resource efficient representation, we look to

determine an explicit function that maps all possible initial states $x_0 \in \mathbb{X}$ to the optimal control action $u^\star(t; x_0)$. The set $\mathbb{X} \subseteq \mathcal{X}$ denotes the working region of the controller and will be mathematically defined later in the paper. For linear models and constraints and quadratic cost functions, the MPC control law can be computed fully offline using multiparametric quadratic programming algorithms, which is often referred to as explicit MPC. However, the extension of explicit MPC to nonlinear and high-dimensional problems has remained an open challenge due to a more complex set of optimality conditions and the potential exponential growth in the number of regions defining the control law as the problem size increases. Therefore, we take the perspective in this work that an approximation to this explicit function will be acceptable for real-time purposes. Deep neural networks (DNNs) have recently become a popular choice for the approximation due to their representational power and the availability of advanced training algorithms in a number of open-source platforms. To construct this data-driven approximation of π_{mpc}, we must first generate a training data set

$$\mathcal{D} = \{(x_1, \pi_{mpc}(x_1)), \ldots, (x_{N_s}, \pi_{mpc}(x_{N_s})\}, \tag{9}$$

where N_s denotes the number of samples. In this case, the OCP (2) needs to be solved for every $x_i \in \mathbb{X}$. A DNN function with fully connected layers can be represented as

$$\mathcal{N}(x; \theta) = \alpha_{L+1} \circ \beta_L \circ \alpha_L \circ \cdots \circ \beta_1 \circ \alpha_1(x), \tag{10}$$

where L and H are the number of hidden layers and number of hidden nodes per layer, respectively. The hidden layers are made up of affine transformations of the output of the previous layer $\alpha_l(\xi_{l-1}) = W_l \xi_{l-1} + b_l$, where $\xi_{l-1} \in \mathbb{R}^H$ for $l = 2, \ldots, L+1$ and $\xi_0 = x$. The functions $\{\beta_l\}_{l=1}^L$ represent nonlinear activation functions that ensure DNNs are universal function approximators under relatively mild conditions Barron (1993). The network parameters $\theta = \{W_1, b_1, \ldots, W_{L+1}, b_{L+1}\}$ then correspond to all the weights and biases that appear in the network.

The DNN parameters are trained by minimizing some loss function – commonly chosen to be the mean squared error (MSE) for regression tasks – evaluated over $\mathcal{D}$:

$$\theta^\star = \underset{\theta}{\operatorname{argmin}} \frac{1}{N_s} \sum_{i=1}^{N_s} \|\pi_{mpc}(x_i) - \mathcal{N}(x_i; \theta)\|^2 \tag{11}$$

which results in the following DNN-based approximate MPC $\pi_{approx}(x) = \mathcal{N}(x; \theta^\star)$ that satisfies

$$\|\pi_{mpc}(x) - \pi_{approx}(x)\| \leq \epsilon_{approx}, \quad \forall x \in \mathbb{X} \tag{12}$$

for some $\epsilon_{approx} > 0$. To generate the training data $\mathcal{D}$, we must select the relevant state space $\mathbb{X}$. Although a natural choice is the feasible region of the MPC law, i.e., $\mathbb{X} = \mathcal{X}_N$, this results in some practical challenges that are important to highlight. First, the determination of the region of attraction for nonlinear systems is computationally intractable and can only be done approximately using outer reachable set approximations, e.g., Althoff et al. (2008). Second, it is unlikely that all states $x \in \mathcal{X}_N$ are equally likely to be observed during closed-loop operation. Therefore, in this work we propose an alternative method based on a *control-oriented* representation of $\mathbb{X}$ defined in terms of the following tube of closed-loop trajectories

$$\mathbb{X} = \textstyle\bigcup_{t=0}^{N_{sim}} X(t), \tag{13}$$

Algorithm 1 Control-oriented generation of training data

Input: f, π_{mpc}, $\mathcal{X}_0$, $\mathcal{W}$, ε, N_{sim}, and $N_{mc} = \frac{N_s}{N_{sim}}$.
Output: Training set $\mathcal{D}$.

```
for i = 1 to N_mc do
    x ← random sample from the set X_0
    for k = 1 to N_sim do
        u ← π_mpc(x)
        D ← D ∪ {(x, u)}
        w ← random sample from the set W
        x ← f(x, u) + w
        if ||x − x_e|| ≤ ε then
            break
```

where $X(t)$ denotes the reachable set of states at time step t and N_{sim} denotes the number of simulation steps. The reachable set is defined recursively as follows

$$X(t+1) = \{f(x(t), \pi_{mpc}(x(t))) \mid x \in X(t)\}, \tag{14}$$

for $t = 0, \ldots, N_{sim} - 1$ given initial states $X(0) = \mathcal{X}_0$. The set $\mathcal{X}_0$ plays an important role in that $\mathbb{X}$ will grow larger in response to larger $\mathcal{X}_0$ – thus, from the perspective of training, it is desired to limit the size of $\mathcal{X}_0$ as much as possible. This strategy also gives us the flexibility to consider both batch (finite time) and continuous operations wherein N_{sim} should be large enough so that the system (approximately) reaches steady state.

We cannot find $\mathbb{X}$ exactly due to the nonlinear dynamics of the system; however, we can generate a collection of N_s random samples in $\mathbb{X}$ using our proposed control-oriented learning procedure summarized in Algorithm 1. A key step in the algorithm is the incorporation of an "artificial" (or simulated) disturbance that is randomly drawn from a set $\mathcal{W} = \{w \mid \|w\| \leq \bar{w}\}$, which accounts for the facts that: (i) π_{approx} is only an approximation to π_{mpc} so the resulting closed-loop trajectories will slightly differ and (ii) unexpected disturbances may occur online that cause our system to deviate from the nominal dynamics (1). The upper bound $\bar{w}$ thus represents a tuning parameter that can be interpreted as a measure of confidence in the nominal model predictions. The closed-loop system is simulated until either final time N_{sim} or the equilibrium condition is reached (within some tolerance ε).

4. KKT-DERIVED ONLINE ADAPTIVE CORRECTION SCHEME WITH PROBABILISTIC VERIFICATION

4.1 Online adaptive correction

The approximation error introduced by the DNN-based MPC law may result in unwanted performance losses. A meaningful performance measure for a control policy $\pi(x)$ is the cost along the closed-loop trajectory.

Definition 4. (Closed-loop performance). Consider the system (1), under some state feedback control policy $u = \pi(x)$. The closed-loop performance for this system over T steps, starting from an initial state $x_0 \in \mathcal{X}_0$ is defined as

$$V_T^\pi(x_0) := \sum_{k=0}^{T-1} \ell\left(x_\pi(k; x_0), \pi(x_\pi(k; x_0))\right), \tag{15}$$

and asymptotic performance corresponds to $T \to \infty$.

The asymptotic performance loss can then be stated as,

$$Loss = \|V_{\infty}^{\pi_{approx}}(x_0) - V_{\infty}^{\pi_{mpc}}(x_0)\|, \tag{16}$$

In order to minimize the asymptotic performance losses due to the approximation errors, it may be desirable to adapt the approximate policy online. However, $V_T^{\pi_{mpc}}(x_0)$ is not available online to quantify the performance loss (16). From Lemma 1, we know that the solution to the steady-state optimization problem (6) is an asymptotically stable equilibrium point of the closed loop system $x(t+1) = f(x(t), \pi_{mpc}(x(t)))$. Hence the asymptotic performance of the closed loop system satisfies the KKT conditions of (6). For any equilibrium point $x = f(x, u)$. the steady-state optimization problem (6) can implicitly be written only in terms of u as

$$\min_{u} \ \tilde{\ell}(u), \quad \text{s.t.} \ \ g(u) \leq 0, \tag{17}$$

and the KKT conditions read as

$$\nabla \tilde{\ell}(u) + \nabla g(u)^{\mathsf{T}} \lambda = 0, \tag{18a}$$

$$g(u) \leq 0, \ \lambda^{\mathsf{T}} g(u) = 0, \ \lambda \geq 0, \tag{18b}$$

where the constraints (6c) are collectively denoted as $g(u)$ and the subset of the active constraints is denoted by $g_{\mathbb{A}}(u) \subseteq g(u)$. Since we do not know the Lagrange multipliers λ, the stationarity condition (18a) can be equivalently stated using the reduced gradient given by $\mathbf{N}^{\mathsf{T}} \nabla \tilde{\ell}(u) = 0$ where $\mathbf{N}$ lies in the nullspace of the active constraint gradients, that is, $\mathbf{N}^{\mathsf{T}} \nabla g_{\mathbb{A}}^{\mathsf{T}}(u) = 0$ (Krishnamoorthy and Skogestad, 2019). Assumption 2 ensures that the active constraint gradient has full row rank and that the nullspace $\mathbf{N}$ is well defined.

Assumption 4. The closed-loop performance using the approximate policy $\pi_{approx}(\cdot)$ converges to some point (x'_e, u'_e) such that $x'_e = f(x'_e, u'_e)$.

Although the notion of convergence in Assumption 4 is central here, it is not guaranteed by design. Validation methods to efficiently ensure probabilistic guarantees that satisfies Assumption 4 will be provided in Section 4.2. When training an approximate policy π_{approx} using the MPC policy π_{mpc}, we want to converge to the same limit point (x_e, u_e), where the KKT conditions (18) hold. However, due to the the approximation error, we have

$$\ell(x_\pi(k; x_0), \pi_{approx}(x_\pi(k; x_0))) \to \ell(x'_e, u'_e) \text{ as } k \to \infty$$

From Lemma 1, the limit point (x'_e, u'_e) obtained using the approximate policy will not be a KKT point of the optimization problem if $(x'_e, u'_e) \neq (x_e, u_e)$ (cf. (7)). Therefore, the deviation from the KKT condition indicates asymptotic performance losses stemming from the MPC policy approximation.

For offset-free asymptotic performance, we can then adapt the approximate control law online to ensure that (x'_e, u'_e) satisfies the KKT conditions (18).

Theorem 2. Let π_{approx} be an approximation of the economic MPC policy π_{mpc}, trained using data collected from Algorithm 1, that satisfies Assumption 4. Offset-free asymptotic performance can then be obtained by using the modified approximate policy

$$u = \pi_{approx}(x) + \delta u =: \tilde{\pi}_{approx}(x), \tag{19}$$

where δu is computed as

$$\dot{\delta u} = -K \begin{bmatrix} g_{\mathbb{A}}(u) \\ \mathbf{N}^{\mathsf{T}} \nabla \tilde{\ell}(u) \end{bmatrix}, \tag{20}$$

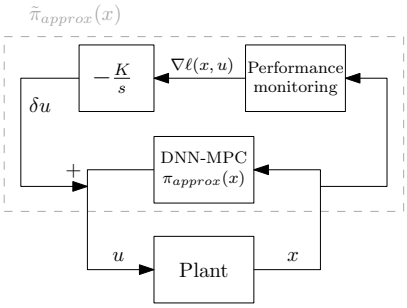

Fig. 1. Schematic of the proposed modified approximate control policy for offset-free asymptotic performance.

Proof Under the modified approximate policy $\tilde{\pi}_{approx}(x)$, the outer loop δu adjusts the control input if the KKT conditions of (17) are not satisfied. The outer loop converges when $\dot{\delta u} = 0$, i.e., $g_{\mathbb{A}}(x, \pi_{approx}(x)) = 0$ and $\mathbf{N}^{\mathsf{T}} \nabla \tilde{\ell}(x, \pi_{approx}(x)) = 0$, which implies that the limit point (x'_e, u'_e) obtained using $\tilde{\pi}_{approx}(x)$ satisfies the KKT conditions of the steady-state optimization problem (6), and hence $(x'_e, u'_e) = (x_e, u_e)$. □

Since δu is based on the KKT conditions of the steady-state optimization problem, the gain K in the modified control policy must be chosen such that it does not significantly affect the dynamics, but adjusts the asymptotic performance. One choice for K is given by $K \propto \begin{bmatrix} \nabla g_{\mathbb{A}}(u) \\ \mathbf{N}^{\mathsf{T}} \nabla^2 \tilde{\ell}(u) \end{bmatrix}^{-1}$ to account for the multivariable interactions. The schematic representation of the proposed scheme is shown in Fig. 1.

4.2 Offline Probabilistic Verification

We assumed the approximate MPC policy π_{approx} ensures convergence to some equilibrium point in Theorem 2, which would hold if we could learn an exact representation, i.e., $\epsilon_{approx} = 0$. However, this is not practically possible for several reasons including a finite amount of training data, local optimization strategies for approximately solving (11), and an *a priori* selected DNN structure that may be of insufficient complexity to exactly represent π_{mpc}.

Instead of trying to enforce Assumption 4 in our training procedure, we instead look to verify it holds by developing a stability test in terms of the following binary performance indicator:

$$\phi(x_0) = \begin{cases} 0 \text{ if closed-loop system converges for } x_0, \\ 1 \text{ otherwise.} \end{cases}$$

The stability test can then be stated mathematically as $\phi(x_0) = 0$ for all $x_0 \in \mathcal{X}_0$, which is equivalently stated as $\max_{x_0 \in \mathcal{X}_0} \phi(x_0) = 0$. Since this is a challenging optimization, we resort to computing a probabilistic estimate of the worst-case performance by associating a probability measure $\Pr_{\mathcal{X}_0}$ over the sample space $\mathcal{X}_0$. We then generate N_v i.i.d. samples of x_0 and define the empirical maximum

$$\hat{E}_{N_v} = \max_{i=1,\ldots,N_v} \phi(x_0^{(i)}). \tag{21}$$

Using seminal results in (Tempo et al., 1997, Theorem 3.1), we can establish that, for any *accuracy* $\epsilon \in (0, 1)$ and *confidence* $\delta \in (0, 1)$, if

$$N_v \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}} \tag{22}$$

then

$$\Pr_{\mathcal{X}_0^{N_v}}\{\Pr_{\mathcal{X}_0}\{\phi(x_0) > \hat{E}_{N_v}\} \leq \epsilon\} \geq 1-\delta \tag{23}$$

This bound is independent of the size of the set $\mathcal{X}_0$ and the probability measure $\Pr_{\mathcal{X}_0}$. Note there are two levels of probabilities since $\hat{E}_{N_v}$ is a random variable that depends on the multisample $\mathbf{x}_0 = \{x_0^{(1)}, \ldots, x_0^{(N_v)}\}$. The scenario bound (23) can be roughly interpreted as ensuring the set of points greater than the estimated worst-case performance has a measure smaller than ϵ, and this is true with probability at least $1-\delta$.

To apply (23) in practice, we must simulate the closed-loop system $x(t+1) = f(x(t), \tilde{\pi}_{approx}(x(t)))$ under N_v randomly selected initial conditions to determine $\hat{E}_{N_v}$. If N_v is chosen to satisfy (22), then $\hat{E}_{N_v} = 0$ and hence all closed-loop trajectories are stable. Given the online adaptive correction scheme, this implies that the approximate control policy $\tilde{\pi}_{approx}$ converges to the optimal equilibrium (x_e, u_e). On the other hand, if any closed-loop trajectory is unstable, then $\hat{E}_{N_v} = 1$ and we have certified that $\tilde{\pi}_{approx}$ does not meet our requirements and we must repeat the training procedure [1] – this may involve a more advanced hyperparameter optimization strategy and/or augmentation of the training data set $\mathcal{D}$. Interested readers are referred to Paulson and Mesbah (2020b) and Krishnamoorthy (2020), respectively, for recent papers that tackle these two important problems.

5. ILLUSTRATIVE EXAMPLE

We use a benchmark Williams-Otto reactor to illustrate the proposed method. Two input streams F_A and F_B are fed to the reactor with pure components A and B, respectively. Useful products P and E are formed by a series of chemical reactions

$$\begin{aligned} A + B &\rightarrow C & k_1 &= 1.6599 \times 10^6 e^{-6666.7/T_r} \\ B + C &\rightarrow P + E & k_2 &= 7.2177 \times 10^8 e^{-8333.3/T_r} \\ C + P &\rightarrow G & k_3 &= 2.6745 \times 10^{12} e^{-11111/T_r} \end{aligned}$$

The process has $n_x = 6$ states and $n_u = 2$ inputs. Feedstream F_A is an unmeasured disturbance and the process is controlled using feedstream F_B and reactor temperature T_r. The stage cost is given by $\ell = -1043.38x_P(F_A+F_B) - 20.92x_E(F_A+F_B) + 79.23F_A + 118.34F_B$, where x_E and x_P denote the mass fractions of components E and P, respectively. The economic MPC problem is setup with a prediction horizon of $N = 60$ samples, with a sampling time of 1 min.

[1] Note that the probabilistic verification can also be performed on the approximate policy π_{approx} to validate that the aapproximate policy converges to *some* equilibrium point (x'_e, u'_e) .

Table 1. Closed-loop performance of the MPC policies π_{mpc}, π_{approx}, and $\tilde{\pi}_{approx}$.

	π_{mpc}	π_{approx}	$\tilde{\pi}_{approx}$
V_T^{π}	-6.2122×10^6	-6.1078×10^6	-6.2083×10^6
Loss	–	1.044×10^5	3877

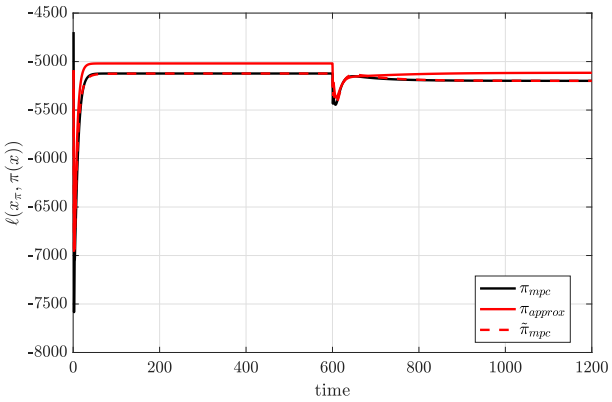

Fig. 2. Closed-loop cost of the MPC policies $\pi_{mpc}(x)$ (solid black), $\pi_{approx}(x)$ (solid red) and the proposed modified approximate policy $\tilde{\pi}_{approx}(x)$ (dashed red).

The training dataset was generated using Algorithm 1 with $N_m = 20$, $N_{sim} = 50$, and $\varepsilon = 10^{-3}$. $\mathcal{X}_0$ was chosen to be a box set with uniform distribution with each state varying between 0 and 0.7. $\mathcal{W}$ was chosen to be a normal distribution with zero mean and a standard deviation of 10^{-2}. The MPC policy π_{mpc} is obtained by solving the optimal control problem (2), which is setup using `CasADi v3.5.1` (Andersson, 2013), and the resulting NLP is solved using `IPOPT` (Wächter and Biegler, 2006). The MPC policy is approximated using a deep neural network with 1 hidden layer with 5 neurons with rectified linear units (ReLU) as the activation function. These hyperparameters were chosen using Bayesian optimization (Snoek et al., 2012). The neural network was then trained using the Levenberg-Marquardt algorithm from the `MATLAB v2019b` deep learning toolbox. For an accuracy of $\epsilon = 0.05$ and confidence $\delta = 0.05$, the closed-loop system was simulated for $N_v = 59$ initial conditions that were randomly generated to ensure that $\hat{E}_{N_v} = 0$.

The process is simulated with the MPC policy $\pi_{mpc}(x)$ (benchmark), the approximate policy $\pi_{approx}(x)$, and the proposed modified approximate policy $\tilde{\pi}_{approx}$. The simulation starts with a feed flowrate $F_A = 1.3$ kg/s, which changes to $F_A = 1.4$ kg/s at $t = 600$. Fig. 2 shows the closed-loop performance with $\pi_{mpc}(x)$ (solid black), $\pi_{approx}(x)$ (solid red), and the proposed modified approximate policy $\tilde{\pi}_{approx}(x)$ (dashed red). As can be seen, the approximate policy π_{approx} leads to a steady-state offset compared to the closed-loop solution obtained with π_{mpc}, whereas the closed-loop performance $\tilde{\pi}_{approx}$ leads to an offset-free asymptotic performance. The closed-loop performance $V_T^{\pi}(x_0)$ for the three different policies are also summarized in Table 1. The modified approximate policy leads to a better performance, since the process is driven to the optimal equilibrium point at steady-state.

6. CONCLUSIONS AND FUTURE WORK

This paper addressed the problem of performance loss due to approximating an economic MPC law using deep neural networks. It was shown that we can account for the approximation errors and reduce performance losses by using a KKT-derived adaptive correction scheme. We also presented an offline probabilistic verification scheme

to establish convergence, which is central to the proposed adaptive correction scheme. An extension of the online correction scheme to handle structural mismatch would entail the use of plant gradients instead of model gradients.

REFERENCES

Althoff, M., Stursberg, O., and Buss, M. (2008). Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 4042–4048.

Andersson, J. (2013). *A General-Purpose Software Framework for Dynamic Optimization*. Ph.D. thesis, Arenberg Doctoral School, KU Leuven.

Angeli, D., Amrit, R., and Rawlings, J.B. (2011). On average performance and stability of economic model predictive control. *IEEE Transactions on Automatic Control*, 57(7), 1615–1626.

Barron, A.R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3), 930–945.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Bertsekas, D.P. (2019). *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA.

Chakrabarty, A., Dinh, V., Corless, M.J., Rundell, A.E., Żak, S.H., and Buzzard, G.T. (2017). Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences. *IEEE Transactions on Automatic Control*, 62(1), 135–148.

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

Csekő, L.H., Kvasnica, M., and Lantos, B. (2015). Explicit MPC-based RBF neural network controller design with discrete-time actual kalman filter for semiactive suspension. *IEEE Transactions on Control Systems Technology*, 23(5), 1736–1753.

Diehl, M., Amrit, R., and Rawlings, J.B. (2010). A Lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control*, 56(3), 703–707.

Drgoňa, J., Picard, D., Kvasnica, M., and Helsen, L. (2018). Approximate model predictive building control via machine learning. *Applied Energy*, 218, 199–216.

Grüne, L. and Pannek, J. (2017). Nonlinear model predictive control. In *Nonlinear Model Predictive Control*, 45–69. Springer.

Hertneck, M., Köhler, J., Trimpe, S., and Allgöwer, F. (2018). Learning an approximate model predictive controller with guarantees. *IEEE Control Systems Letters*, 2(3), 543–548.

Hirose, N., Tajima, R., and Sukigara, K. (2018). MPC policy learning using DNN for human following control without collision. *Advanced Robotics*, 32(3), 148–159.

Karg, B. and Lucia, S. (2020). Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Transactions on Cybernetics*, 50(9), 3866–3878.

Krishnamoorthy, D. (2020). Sensitivity-based data augmentation for learning an approximate model predictive controller. *arXiv preprint arXiv:2009.07398*.

Krishnamoorthy, D. and Skogestad, S. (2019). Online process optimization with active constraint set changes using simple control structures. *Industrial & Engineering Chemistry Research*, 58(30), 13555–13567.

Paulson, J.A. and Mesbah, A. (2020a). Approximate closed-loop robust model predictive control with guaranteed stability and constraint satisfaction. *IEEE Control Systems Letters*, 4(3), 719–724.

Paulson, J.A. and Mesbah, A. (2020b). Data-driven scenario optimization for automated controller tuning with probabilistic performance guarantees. *IEEE Control Systems Letters*, 5(4), 1477–1482.

Snoek, J., Larochelle, H., and Adams, R.P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2951–2959.

Tempo, R., Bai, E.W., and Dabbene, F. (1997). Probabilistic robustness analysis: explicit bounds for the minimum number of samples. *Systems and Control Letters*, 30, 237–242.

Tøndel, P., Johansen, T.A., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica*, 39(3), 489–497.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57.

Zhang, X., Bujarbaruah, M., and Borrelli, F. (2019). Safe and near-optimal policy learning for model predictive control using primal-dual neural networks. In *Proceedings of the American Control Conference*, 354–359.

Appendix A. PROOF OF THEOREM 1

Let $x^\star(\cdot|t)$ and $u^\star(\cdot|t)$ denote the optimal state and input sequence for (2) at time t, respectively. We define the following feasible candidate sequences

$$\tilde{x}(\cdot|t+1) = \{x^\star(1|t), \cdots, x^\star(N|t), x_e\}$$
$$\tilde{u}(\cdot|t+1) = \{u^\star(1|t), \cdots, u^\star(N-1|t), u_e\},$$

since (x_e, u_e) is assumed to be a feasible equilibrium pair and $x^\star(N|t) = x_e$ from (2d). We must first establish a bound on the optimal cost difference to use the Lyapunov stability framework

$$\begin{aligned}
&\tilde{V}(x(t+1)) - \tilde{V}(x(t)) \\
&\leq \sum_{k=0}^{N-1} L(\tilde{x}(k|t+1), \tilde{u}(k|t+1)) - \sum_{k=0}^{N-1} L(x^\star(k|t), u^\star(k|t)) \\
&\leq L(x_e, u_e) - L(\tilde{x}_0, \tilde{u}_0).
\end{aligned}$$

Now, from (5) and (8), we can derive

$$\begin{aligned}
&\tilde{V}(x(t+1)) - \tilde{V}(x(t)) \\
&\leq \ell(x_e, u_e) + \lambda(x_e) - \lambda(f(x_e, u_e)) \\
&\qquad - \ell(x_0, u_0) - \lambda(x_0) + \lambda(f(x_0, u_0)) \\
&\leq -\alpha(\|x - x_e\|).
\end{aligned}$$

This implies $\tilde{V}(x)$ is a Lyapunov function for the nominal closed-loop system (with recursive feasibility guaranteed by the terminal equilibrium constraint) such that the stability claim immediately follows.