2025 23rd European Control Conference (ECC)
June 24-27, 2025. Thessaloniki, Greece

Hybrid Learning for Model Predictive Control Approximation

Christos N. Mavridis*, and Karl Henrik Johansson*

Abstract— **We study the problem of approximating a model predictive controller (MPC) with learning models to facilitate real-time operation. In particular, we investigate how the use of a hybrid learning model can tighten the statistical learning bounds used for stability guarantees given by existing robust data-driven MPC approaches. We propose a hybrid learning framework with a finite set of state-dependent modes, each consisting of a supervised regression model. The mode-switching signal corresponds to a state space partition produced by solving a homotopy optimization problem that implicitly minimizes the Lipschitz constant of the regression model in each mode. The cardinality of the partition is decided by a bifurcation phenomenon, inducing a performance-complexity trade-off that is discussed. The proposed MPC approximation framework is validated on a nonlinear benchmark problem.**

I. Introduction

Data-driven methods for system analysis and control have gained significant popularity in recent years. However, the adoption of these methods in control problems is not yet attainable due to the lack of theoretical guarantees regarding closed-loop stability, which are strongly dependent on the data and the behavior of the learning systems used for approximation [1].

A number of methods towards robustness and stability guarantees of data-driven control approaches have already been developed [1]–[5]. In particular, model predictive control (MPC) is a well-suited control framework to meet these goals, as it can handle nonlinear system dynamics, hard constraints, and performance criteria [6]. MPC relies on repeatedly solving an optimal control problem based on predicted system trajectories. However, the complexity of this repeated process for every timestep, induces a computational cost that often renders real-time operation challenging [6]. Therefore, approximating the resource-intensive MPC scheme with a data-driven approximation of its control law has received increasing attention. A few methods have been developed to generate such approximations, including pre-calculation of the control law in explicit MPC [7], and learning frameworks based on supervised and reinforcement learning [8]–[11].

However, the application of an approximate MPC controller via data-driven regression models in a closed-loop system can greatly amplify the learning errors [12]. Although principles from robust control can be applied in the presence of approximation error [3], the fundamental limitation remains in the quantification of its bounds [13]. The bound for the closed-loop error depends directly on the regression error within the training dataset, the generalization properties of the learning system outside the training dataset, and the properties of the functions of the controller and the learning model, usually quantified in the form of Lipschitz constants. In that sense, the well-known problem of over-fitting versus generalization is present [14]. Decreasing the regression error requires an increase in the complexity of the learning model, which induces poor generalization, quantified by increased approximation error outside the training dataset and often unnecessarily loss of smoothness, measured by an increased Lipschitz constant for the regression model.

A. Contribution.

In this work, we propose a hybrid learning architecture to approximate a model-predictive controller. Hybrid systems, described by interacting continuous and discrete dynamics, are a powerful modeling tool to approximate highly nonlinear dynamics by a collection of simpler models, enhancing model explainability and robustness [15]–[18].

In particular, we develop a hybrid learning framework with a finite set of state-dependent modes, each consisting of a supervised regression model. There are no restrictions on the regression model used, apart from Lipschitz continuity. The mode-switching signal corresponds to a state space partition produced by solving a homotopy optimization problem using the Online Deterministic Annealing (ODA) approach [19], [20]. We show that this process implicitly minimizes the Lipschitz constant of the regression model in each mode. In addition, the cardinality of the partition is decided by a bifurcation phenomenon that depends on the underlying statistics of the training dataset [21].

We show that learning with hybrid systems can reduce the approximation error and the Lipschitz constant of the local models (compared to a single regression model), while providing a mechanism to determine regions of the state space where more observations are needed. We discuss the performance-complexity trade-off induced by the growing cardinality of the number of modes by the optimization approach. Finally, the proposed MPC approximation framework is validated on a nonlinear benchmark problem.

B. Structure.

In Section II we introduce the problem formulation and the available bounds for stability guarantees for learning-based MPC approximation. In Section III we develop the hybrid learning framework for MPC approximation and formally describe its properties. Finally, in Section IV we validate our approach on a nonlinear benchmark problem.

*Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm. emails:{mavridis,kallej}@kth.se.

Research partially supported by the Swedish Foundation for Strategic Research (SSF) grant IPD23-0019.

C. Notation.

The sets $\mathbb{R}$ and $\mathbb{Z}$ represent the sets of real and integer numbers, respectively, while $\mathbb{Z}_+$ represents the set of non-negative integers. For a real matrix $A \in \mathbb{R}^{n\times m}$, $A^{\mathrm{T}} \in \mathbb{R}^{m\times n}$ denotes its transpose and $\mathrm{vec}(A) \in \mathbb{R}^{mn}$ the vectorization of A. The $n \times n$ identity matrix is denoted I_n. $A \succeq 0$ is a positive semi-definite matrix, and the condition $A \succeq B$ is understood as $A - B \succeq 0$. Unless otherwise specified, random variables $\mathcal{X} : \Omega \to \mathbb{R}^d$ are defined in a probability space $(\Omega, \mathbb{F}, \mathbb{P})$. The probability of an event is denoted $\mathbb{P}[\mathcal{X} \in S] := \mathbb{P}[\omega \in \Omega : \mathcal{X}(\omega) \in S]$, and the expectation operator $\mathbb{E}[\mathcal{X}] = \int_\Omega \mathcal{X} \mathrm{d}\mathbb{P}$. In case of multiple random variables $(\mathcal{X}, \mathcal{Y})$ and a deterministic function f, the expectation operator $\mathbb{E}[f(\mathcal{X}, \mathcal{Y})]$ is understood with respect to the joint probability measure, while $\mathbb{E}[\mathcal{X}|\mathcal{Y}] := \mathbb{E}[\mathcal{X}|\sigma(\mathcal{Y})]$ denotes the expectation of $\mathcal{X}$ conditioned to the σ-field of $\mathcal{Y}$. Stochastic processes $\{\mathcal{X}(k)\}_k$, $k \in \mathbb{Z}_+$, are defined in the filtered probability space $(\Omega, \mathbb{F}, \{\mathcal{F}_n\}_n, \mathbb{P})$, where $\mathcal{F}_n = \sigma(\mathcal{X}(k)|k \le n)$, $k \in \mathbb{Z}_+$, is the natural filtration. The indicator function of the event $[\mathcal{X} \in S]$ is denoted $\mathbb{1}_{[\mathcal{X}\in S]}$ and $\otimes$ denotes the Kronecker product. Finally, "min" defines the minimization operator while "minimize" defines a minimization problem.

II. Problem Formulation

Consider a discretized nonlinear system with no disturbances of the form:

$$x_{t+1} = f(x_t, u_t), \quad t \in \mathbb{Z}_+, \tag{1}$$

where $\{t\}_{t\in\mathbb{Z}_+}$ is the time sequence, $x_t \in \mathcal{X} \subseteq \mathbb{R}^d$ the state vector at time t, $u_t \in \mathcal{U} \subseteq \mathbb{R}^r$ the input at time t, and $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ define the dynamics. An MPC controller $\{u_t\}_{t\in\mathbb{Z}_+}$ for system (1) with a given objective $J : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_+$, time horizon T, and terminal cost $V : \mathcal{X} \to \mathbb{R}_+$, is computed by repeatedly solving the following optimal control problem for every time t:

$$\begin{aligned} \underset{\{u_k\}}{\text{minimize}} \quad & \sum_{k=0}^{T-1} J(x_{t+k}, u_{t+k}) + V(x_{t+T}) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k), \\ & x_k \in \mathcal{X}_k, \\ & u_k \in \mathcal{U}_k, \\ & \forall k \in \{0, \dots, T\}. \end{aligned} \tag{2}$$

This framework is typically computationally intensive, as a nonlinear optimization problem is solved at each step $t \in \mathbb{Z}_+$. To bypass this limitation, by representing the implicit feedback control law:

$$u_t^* = \kappa(x_t), \tag{3}$$

where u_t^* is the solution of (2) at time t, one can seek to approximate the mapping $\kappa : \mathcal{X} \to \mathcal{U}$ by a regression model $\tilde{\kappa} : \mathcal{X} \times \Theta \to \mathcal{U}$, such that:

$$\kappa(x_t) = \tilde{\kappa}(x_t, \theta) + e_t, \quad t \in \mathbb{Z}_+, \tag{4}$$

where $\theta \in \Theta \subseteq \mathbb{R}^{d_\theta}$ is the parameter vector for the regression model, and $\{e_t\}_t$ is the induced approximation error sequence. The approximate feedback controller $\tilde{\kappa}(x, \theta)$ can then be used in the closed-loop system, as the forward pass of the regression model is orders of magnitude faster than solving problem (2).

A. Approximation error bounds.

The introduction of an approximate controller $\tilde{\kappa}(x, \theta)$ results in loss of any performance guarantees given by the MPC framework. To provide any potential guarantees on the performance of the closed loop system, there is a need to determine a bound on the approximation error e_t in (4). In the following, we introduce a standard approach to bound the approximation error e_t. Variations of this approach are standard in the related literature (see, e.g., [3]). First, Assumptions 1 and 2 are Lipschitz conditions aiming to bound the fluctuation rate of the functions $\kappa(x)$ and $\tilde{\kappa}(x, \theta)$.

Assumption 1: The MPC control law defined in (3) is Lipschitz continuous, i.e., there exists $L_\kappa > 0$ such that:

$$\|\kappa(x_\mathrm{a}) - \kappa(x_\mathrm{b})\| \le L_\kappa \|x_\mathrm{a} - x_\mathrm{b}\|, \tag{5}$$

for all $x_\mathrm{a}, x_\mathrm{b} \in \mathcal{X}$.

Assumption 2: The regression model defined in (4) is Lipschitz continuous, i.e., for a any given θ, there exists $L_{\tilde{\kappa}} > 0$ such that:

$$\|\tilde{\kappa}(x_\mathrm{a}, \theta) - \tilde{\kappa}(x_\mathrm{b}, \theta)\| \le L_{\tilde{\kappa}} \|x_\mathrm{a} - x_\mathrm{b}\|, \tag{6}$$

for all $x_\mathrm{a}, x_\mathrm{b} \in \mathcal{X}$.

In addition, Assumptions 3, 4 are made on the training dataset and learning algorithm and aim to bound the approximation error.

Assumption 3: Given a training dataset D of pairs $\{(x_i, \kappa(x_i))\}_{x_i\in D}$, there exists an attainable solution θ^* for the parameters of the regression model $\tilde{\kappa}(x, \theta)$, such that:

$$\|\kappa(x_i, \theta^*) - \tilde{\kappa}(x_i, \theta^*)\| \le \epsilon_D, \quad \forall x_i \in D, \tag{7}$$

for an arbitrary small $\epsilon_D > 0$.

Assumption 4: The training dataset $D = \{(x_i, \kappa(x_i))\}$ is dense enough within $\mathcal{X} \times \mathcal{U}$, such that there exists $\delta > 0$ for which:

$$\|x - x_i\| \le \delta, \quad \forall x_i \in D, \ \forall x \in \mathcal{X}. \tag{8}$$

Given Assumptions 1, 2, 3, 4, we can bound the approximation error e_t in (4) using the results of Theorem 1.

Theorem 1: If Assumptions 1, 2, 3, 4, hold, then the approximation error

$$e(x) = \kappa(x) - \tilde{\kappa}(x, \theta^*), \quad x \in \mathcal{X}, \tag{9}$$

is bounded by

$$\|e(x)\| \le (L_\kappa + L_{\tilde{\kappa}})\delta + \epsilon_D, \quad \forall x \in \mathcal{X}. \tag{10}$$

Proof: Choose any $x \in \mathcal{X}$ and find the representing training vector x_i such that

$$x_i = \arg\min_{x_j} \|x - x_j\|, \quad x_j \in D. \tag{11}$$

Then by Assumptions 1, 2, 3, and the triangular inequality we get:

$$\begin{aligned}\|\kappa(x)-\tilde{\kappa}(x,\theta^*)\| =& \|\kappa(x)-\kappa(x_i)+\\ & \tilde{\kappa}(x,\theta^*)-\tilde{\kappa}(x_i,\theta^*)+\\ & -\kappa(x_i)+\tilde{\kappa}(x_i,\theta^*)\|\\ \leq& \|\kappa(x)-\kappa(x_i)\|+\\ & \|\tilde{\kappa}(x,\theta^*)-\tilde{\kappa}(x_i,\theta^*)\|+\\ & \|\kappa(x_i)-\tilde{\kappa}(x_i,\theta^*)\|\\ =& (L_\kappa+L_{\tilde{\kappa}})\|x-x_i\|+\epsilon_D\end{aligned} \tag{12}$$

Finally, by Assumption 4, it holds that $\|x-x_i\|\leq\delta$, which completes the proof. ■

Theorem 1 gives a bound to the approximation error $\|\epsilon_t\|$ that depends on the Lipschitz constant of the dynamics L_κ, as well as the training dataset (through δ), the training algorithm (through ϵ_D), and complexity of the training model (through $L_{\tilde{\kappa}}$). The knowledge of this bound for the approximate closed-loop system:

$$\begin{aligned}x_{t+1} &= f(x_t,\tilde{\kappa}(x_t,\theta^*))\\ &= f(x_t,\kappa(x_t,\theta^*)+\epsilon_t),\quad t\in\mathbb{Z}_+,\end{aligned} \tag{13}$$

allows for robust control approaches that guarantee boundedness of the system to be used [3].

However, it is important to note that the fundamental limitation of such results is in the actual values of the parameters L_κ, $L_{\tilde{\kappa}}$, δ, and ϵ_D. In Section III, we introduce a hybrid learning approach to reduce $L_{\tilde{\kappa}}$, and ϵ_D, and provide a constructive way to inform the system on the locations of new training data that, if observed, will reduce δ, as well.

III. Learning with Hybrid Systems

A general discrete-time autonomous hybrid system can be written in the form:

$$\begin{cases}\chi_t &= g_{\sigma_t}(\chi_t,v_t)\\ \sigma_t &= \phi(\sigma_t,\chi_t,v_t)\end{cases},\quad t\in\mathbb{Z}_+, \tag{14}$$

where $\chi_t\in S\subseteq\mathcal{U}\subseteq\mathbb{R}^r$, $v_t\in\mathbb{R}^d$, $\sigma_t\in\{1,\dots,s\}$, $g_i:\mathbb{R}^r\times\mathbb{R}^d\to\mathbb{R}^r$, $\forall i\in\{1,\dots,s\}$, and $\phi:\{1,\dots,s\}\times\mathbb{R}^r\times\mathbb{R}^d\to\{1,\dots,s\}$. The signal σ_t is called the mode-switching signal and can take a finite number of s values that are called the modes of the hybrid system.

We are interested in input-output regression models. Therefore, we will work with g_{σ_t} that do not depend on χ_t. Moreover, we will focus on hybrid systems with state-dependent modes. In particular, we will assume that ϕ is given by a polyhedral partition rule of the form:

$$\sigma_t=i\Leftrightarrow\chi_t\in S_i,\quad t\in\mathbb{Z}_+, \tag{15}$$

where $\{S_i\}_{i=1}^s$ define a polyhedral partition of S, i.e., $S_i\subset S$ is a polyhedron in $\mathbb{R}^r$ for every i, $S_i\cap S_j=\emptyset$ for $i\neq j$, and $\bigcup_i S_i=S$. A hybrid regression model $\tilde{\kappa}(x_t)$ to approximate the MPC scheme $\kappa(x_t)$ then takes the form:

$$\begin{cases}\tilde{\kappa}(x_t) &= g_{\sigma_t}(x_t,\theta_{\sigma_t})\\ \sigma(x_t) &= \sum_{i=1}^s \mathbb{1}_{[x_t\in S_i]}\cdot i\end{cases},\quad t\in\mathbb{Z}_+, \tag{16}$$

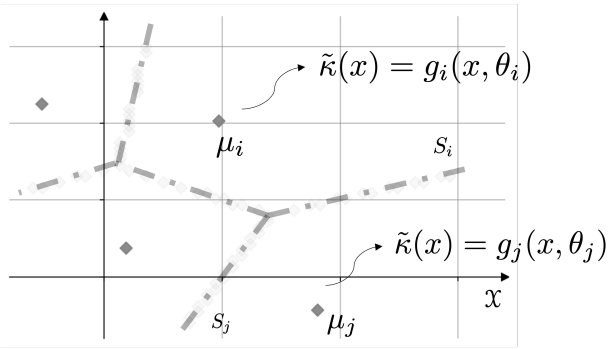

Fig. 1: Illustration of the state-dependent switching regression model in (16). The partition $\{S_i\}_{i=1}^s$ of the state space S is a polyhedral Voronoi partition induced by the parameters $\{\mu_i\}_{i=1}^s$ and a Bregman divergence d_ρ.

where the parameters to be identified are the partition $\{S_i\}$, the model parameter functions $\{g_i\}$ and vectors $\{\theta_i\}$ for each mode $i\in\{1,\dots,s\}$, as well as the number of modes s. A visual representation of the hybrid learning model (16) is shown in Fig. 1.

A. The learning problem.

The general identification problem for a hybrid learning system of the form (16) can be formulated as a stochastic optimization problem [16]–[18] over the parameters $\{s,\{\theta_i\}_{i=1}^s,\{S_i\}_{i=1}^s\}$, as follows:

$$\underset{s,\{\theta_i\},\{S_i\}}{\text{minimize}}\ \mathbb{E}\left[\sum_{i=1}^s \mathbb{1}_{[X\in S_i]}d_\rho\left(\kappa(X),g_i(X,\theta_i)\right)\right], \tag{17}$$

where $\{g_i\}$ are given parametric models, and $X\in\mathbb{S}$ represents a random variable with realizations x_i in the training dataset D. The nonnegative measure d_ρ is an appropriately defined dissimilarity measure. It is clear that the optimization problem (17) is computationally hard and becomes intractable as the number of modes and the state dimension increases. In particular, the number of modes s is unknown and completely alters the cardinality and the domain of the set of parameter vectors $\{\theta_i\}_{i=1}^s$ that represent the dynamics of the system. In addition, a parametric representation for the polyhedral regions $\{S_i\}$ should be defined.

To represent the regions $\{S_i\}$, we will follow a Voronoi tessellation approach based on prototypes [19], [20]. We introduce a set of parameters $\mu:=\{\mu_i\}_{i=1}^s$, $\mu_i\in S$, where s is to be identified. Next we define the regions $\{S_i\}$ by:

$$S_i=\left\{x\in S: i=\arg\min_j d_\rho(x,\mu_j)\right\},\ i=1,\dots,s. \tag{18}$$

The measure d_ρ is designed as a Bregman divergence such that the Voronoi regions Σ_i are polyhedral. Bregman divergences is a family of dissimilarity measures that includes the squared Euclidean distance and the Kullback-Leibler divergence. In this work, the squared Euclidean distance is used. For more information, the reader is referred to [20], [22]. An illustration is provided in Fig. 1.

Problem (17) can be approximated by two stochastic optimization problems. Assuming the number of modes s and the partition $\{S_i\}_{i=1}^s$ are known, the optimization problem

$$\underset{\{\theta_i\}}{\text{minimize}} \ \mathbb{E}\left[\sum_{i=1}^{s} \mathbb{1}_{[X\in S_i]} d_\rho\left(\kappa(X), g_i(X,\theta_i)\right)\right] \tag{19}$$

is a regression problem for each mode of the system that can be solved with stochastic optimization methods (e.g., stochastic gradient descent or accelerated optimization methods, in the case of artificial neural network models g_i).

Assuming $\{\theta_i\}_{i=1}^s$ are known, we seek to find an optimal partition of the state space subject to two main criteria. First, we need to facilitate reduced approximation error for each learning model in (19). Second, we need to minimize the Lipschitz constant L_{g_i} for each mode $i = 1, \ldots, s$. To accommodate for both criteria, we introduce the augmented vector:

$$\Psi = \begin{bmatrix} X \\ \kappa(X) \end{bmatrix} \in \Sigma \subseteq \mathbb{R}^{r+d}, \tag{20}$$

and a set of augmented prototype vectors $w := \{w_i\}_{i=1}^s$, $w_i \in \Sigma$, $i = 1, \ldots, s$:

$$w_i = \begin{bmatrix} \mu_i \\ \xi_i \end{bmatrix} \in \Sigma, \ i = 1, \ldots, s. \tag{21}$$

Then we solve the optimization problem:

$$\underset{w}{\text{minimize}} \ \mathbb{E}\left[\sum_{i=1}^{s} \mathbb{1}_{[\Psi\in\Sigma_i(w)]} d_\rho\left(\Psi, w_i\right)\right]. \tag{22}$$

The clustering problem (22) finds the optimal prototypes $\{w_i\}_{i=1}^s$ that define a polyhedral partition $\{\Sigma_i\}_{i=1}^s$ of Σ, subject to the joint distribution of $(X, \kappa(X))$. At the same time, the prototypes $\{\mu_i\}_{i=1}^s$ within the vectors $\{w_i\}$ define a polyhedral partition of S. Notice that (22) is designed to construct an optimal partition in the space of $\Psi = (X, \kappa(X))$, which is equivalent to a piece-wise constant approximation of the function $\kappa(X)$ (defined over the partition $\{S_i\}$). In other words, solving (22), implies that the partition $\{S_i\}$ is optimal with respect to constant function models $g_i = c_i$, $i = 1, \ldots, s$. Therefore, the partition $\{S_i\}$ is optimal for regression models with zero Lipschitz constant, which means that the regions S_i are designed, according to the joint distribution of $(X, \kappa(X))$, such that each regression model used has minimum Lipschitz constant L_{g_i}. Thus, solving (22) implicitly minimizes $\{L_{g_i}\}$.

B. Homotopy optimization.

To solve the optimization problem (22) we use a homotopy optimization approach. We adopt the Online Deterministic Annealing (ODA) method [19], [20], that also addresses the question of finding the optimal number of modes s according to a performance-complexity trade-off. The key idea is to define a probability space over an arbitrary number of codevectors, while constraining their distribution using a maximum-entropy principle at different levels.

First we define a quantizer $Q : \Sigma \to \Sigma$ as a stochastic mapping of the form:

$$Q(x) = w_i \ \text{ with probability } p(w_i|x). \tag{23}$$

Then we formulate the multi-objective optimization problem

$$\underset{w}{\text{minimize}} \ F_\lambda(w) = (1-\lambda)D(w) - \lambda H(w), \ \lambda \in [0, 1). \tag{24}$$

The term

$$D(w) = \mathbb{E}\left[d\left(X, Q\right)\right] = \int p(x) \sum_i p(w_i|x) d_\rho(x, w_i) \ \mathrm{d}x \tag{25}$$

is a generalization of the objective in (22), and

$$\begin{aligned} H(w) &= \mathbb{E}\left[-\log P(X, Q)\right] \\ &= H(X) - \int p(x) \sum_i p(w_i|x) \log p(w_i|x) \ \mathrm{d}x \end{aligned} \tag{26}$$

is the Shannon entropy. This is now a problem of finding the locations $\{w_i\}$ and the corresponding probabilities $\{p(w_i|x) = \mathbb{P}[Q = w_i | X = x]\}$. Notice that, for $p(w_i|x) = \mathbb{1}_{[\phi\in\Sigma_i(w)]}$ and $\lambda = 0$, (24) is equivalent to (22). In that sense, (24) introduces extra optimization parameters in the probabilities $\{p(w_i|x)\}$, and the parameter λ that defines a homotopy F_λ. However, the use of the conditional probabilities $\{p(w_i|x)\}$ allows for the definition of the entropy term H, which introduces several useful properties [16], [19]–[21], [23], [24].

The main idea of the ODA approach is to solve a sequence of optimization problems of the form (24) with decreasing values of λ. This process then becomes a homotopy optimization method [25]. In particular, reducing the values of λ defines a direction that resembles an annealing process [20]. This process introduces robustness with respect to initial conditions [20], [26]. Moreover, it induces a bifurcation phenomenon, with respect to which, the number of unique codevectors $s(\lambda)$ increases as λ decreases. Mathematically, this occurs when the existing solution w^* for (24) is no longer the minimum of the free energy, as the temperature λ crosses a critical value [19], [21].

In practice, we can detect the bifurcation points by introducing perturbing pairs of codevectors at each temperature level λ. The perturbed codevectors will merge with their pair if a critical temperature has not been reached and separate otherwise [19]–[21]. The solution to (24) can be computed using a gradient-free stochastic approximation algorithm, given in Theorem 2.

Theorem 2 ([19]): The sequence $w_i(t)$ constructed by the recursive updates

$$\begin{cases} \rho_i(t+1) &= \rho_i(t) + \beta(t)\left[\hat{p}_i(t) - \rho_i(t)\right] \\ \sigma_i(t+1) &= \sigma_i(t) + \beta(t)\left[\psi_t \hat{p}_i(t) - \sigma_i(t)\right], \end{cases} \tag{27}$$

where ψ_t represents external input with $\sum_t \beta(t) = \infty$, $\sum_t \beta^2(t) < \infty$, and the quantities $\hat{p}_i(t)$ and $w_i(t)$ are recursively updated as follows:

$$w_i(t) = \frac{\sigma_i(t)}{\rho_i(t)}, \quad \hat{p}_i(t) = \frac{\rho_i(t) e^{-\frac{1-\lambda}{\lambda} d_\rho(x_t, w_i(t))}}{\sum_i \rho_i(t) e^{-\frac{1-\lambda}{\lambda} d_\rho(x_t, w_i(t))}}, \tag{28}$$

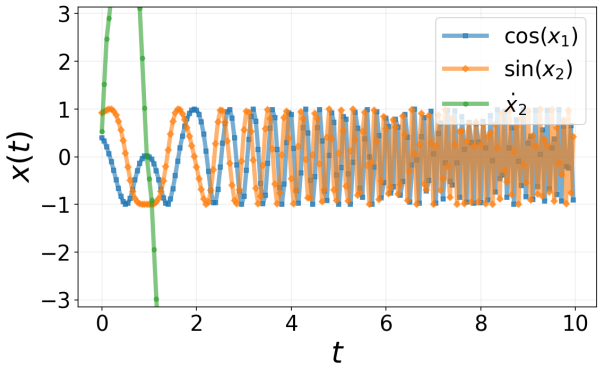

Fig. 2: Closed-loop system evolution of (32) using approximate MPC with a neural network regression model.

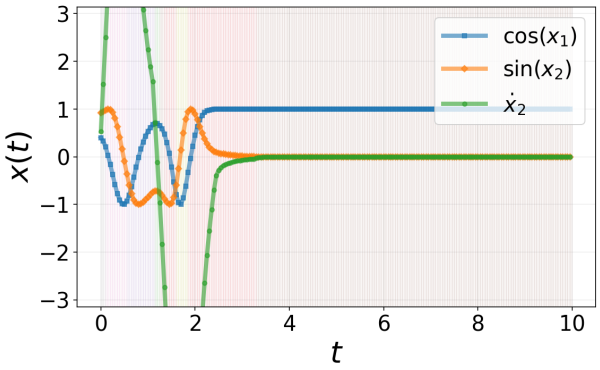

Fig. 3: Closed-loop system evolution of (32) using approximate MPC with a hybrid learning model of the form (16). The mode-switching behavior is shown.

converges almost surely to a solution of (24).

Notice that the dynamics of (27) can be expressed as:

$$\Delta w_i(t+1) = \frac{\beta(t)}{\rho_i(t)}\left[\frac{\sigma_i(t+1)}{\rho_i(t+1)}\left(\rho_i(t) - \hat{p}_i(t)\right) + \psi_t \hat{p}_i(t) - \sigma_i(t)\right], \tag{29}$$

where the recursive updates take place for every codevector w_i sequentially. This is a discrete-time dynamical system that presents bifurcation phenomena with respect to the parameter λ, i.e., the number of equilibria of this system changes with respect to the value λ which is hidden inside the term $\hat{p}_i(t)$ in (28). According to this phenomenon, the number of distinct values of w_i is finite, and the updates need only be taken with respect to these values that we call "effective codevectors".

This is the number of modes used for our hybrid learning model (16). There is a trade-off between performance and complexity as the value of λ reduces and the number of modes s increases. In this work, we use a termination criterion and keep the minimum value λ^* such that a predefined minimum number of training samples x_i are associated to each region S_i, $i = 1, \ldots, s$.

C. Mitigating the jumping effect.

To mitigate the jumping behavior of the controller that may increase the closed-loop error, one can make use of the association probabilities

$$p(x_i|x_t) = \frac{e^{-\frac{1-l}{l} d_\rho(x_t, x_i)}}{\sum_j e^{-\frac{1-l}{l} d_\rho(x_t, x_i)}}, \tag{30}$$

for a constant $l > 0$. Thus, a weighted regression model is constructed:

$$\tilde{\kappa}(x_t) = \sum_{i=1}^{s} p(x_i|x_t) g_i(x_t, \theta_i). \tag{31}$$

IV. Simulation Results

We evaluate the proposed hybrid learning approach for approximate MPC on a discretized version of the actuated inverted pendulum system, given by the non-linear dynamics:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{u}{ml^2} - \frac{g}{l}\sin(x_1 + \pi), \end{aligned} \tag{32}$$

where, x_1 represents the angle of the pendulum at the downward position, $x_2 = \dot{x}_1$ represents the angular velocity, and u is the input torque. The parameters are chosen such that $m = 1$ kg is the mass of the pendulum, $g = 9.81$ m/s^2 is the acceleration due to gravity, and $l = 1$ m is the length of the pendulum. The state space is defined by $-2\pi \leq x_1 \leq 2\pi$ and $-1 \leq x_2 \leq 1$.

The objective of the MPC is to stabilize the upper position of the pendulum. A quadratic cost $J(x_t, u_t) = x_t^{\mathrm{T}} Q x_t + u_t^{\mathrm{T}} R u_t$ and terminal cost $V(x_T) = x_T^{\mathrm{T}} P x_T$, are designed, with weights $Q = P = \mathrm{diag}(10, 1)$ and $R = 0.1$, where diag denotes a diagonal matrix.

A dataset D consisting of $|D| = 2000$ randomly selected samples in the state space was selected. The proposed hybrid learning approach is compared to a standard regression learning approach using fully connected artificial neural networks with $l = 2$ hidden layers of $n = 10, 10$ neurons with activation function $\sigma_\alpha = \tanh$, trained using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [27]. The hybrid system is of the form (16) with g_i given by an identical neural network architecture as above. Our metrics include the average approximation error $\bar{e} = \mathbb{E}[e_t]$ on the training dataset D, the maximum approximation error e_D on D, and the Lipschitz constant of the regression model $L_{\tilde{\kappa}}$, which are directly related to the closed-loop system guarantees through Theorem 1.

The closed-loop system evolution for a randomly selected initial condition $x_0 = (1.16, 0.53)$ using the two different

Model	Region	$\bar{e}$	e_D	L_κ	$L_{\tilde{\kappa}}$
NN	S	.13	3.96	324.62	354.19
Hybrid	S_1	.0013	.039	114.26	113.92
	S_2	.037	.35	67.35	76.76
	S_3	.058	2.42	246.65	326.42
	S_4	.028	.20	283.65	285.59
	S_5	.0076	.13	61.11	61.62
	S_6	.083	2.81	161.71	181.29

TABLE I: The values of $\bar{e}$, e_D, L_κ, and $L_{\tilde{\kappa}}$ for the closed-loop system (32) using different regression models for approximate MPC. The value of these values is proportional to the approximation error bound through Theorem 1.

approaches for MPC approximate control is shown in Fig. 2 and Fig. 3. The values of the parameters $\bar{e}$, e_D, $L_{\tilde{\kappa}}$, as well as the Lipschitz constant L_κ are shown in Table I. The Lipschitz constants are numerically approximated and constitute underestimates of the actual minimal constants.

A. Discussion

While in both cases, Theorem 1 can be used to mathematically establish boundedness conditions for the closed-loop system, it is apparent in Fig. 2 and 3 that the actual value of the bounds is crucial, as the neural network model fails to provide a stabilizing controller, while the hybrid learning approach succeeds. A closer examination of the parameters in Table I, reveals that the use of a hybrid learning algorithm of the form (16) trained using (19) and (22), can lower the values of $\bar{e}$, e_D, L_κ, and $L_{\tilde{\kappa}}$. A partition of the state space is created such that each model g_i can be trained with lower approximation error e_D and increased smoothness, quantified by the Lipschitz constant $L_{\tilde{\kappa}}$. The latter is acquired by the optimization problem (22) that finds the optimal partition such that in every given region S_i, a constant function approximation model (zero Lipschitz constant) is optimized.

Finally, the partition of the state space given by $\{S_i\}$ can be used to determine the location of new data that could help improve the performance of the controller. In particular, observe that in every region S_i, the values of e_D and $L_{\tilde{\kappa}}$ are decreased compared to the neural network model. However, the difference for S_3 and S_4 is not as significant. This information can be used to require more data within these regions, thus further reducing e_D, $L_{\tilde{\kappa}}$, as well as δ.

V. Conclusion

We studied the effect of a hybrid learning model in tightening the statistical learning bounds used for stability guarantees given by existing robust data-driven MPC approaches. We proposed a hybrid learning framework with a finite set of state-dependent modes, each consisting of a supervised regression model. The mode-switching signal corresponds to a state space partition produced by solving a homotopy optimization problem that implicitly minimizes the Lipschitz constant of the regression model in each mode. The cardinality of the partition is decided by a bifurcation phenomenon, inducing a performance-complexity trade-off. The proposed MPC approximation framework is validated on a nonlinear benchmark problem.

References

[1] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[2] B. M. Åkesson and H. T. Toivonen, "A neural network model predictive controller," *Journal of Process Control*, vol. 16, no. 9, pp. 937–946, 2006.

[3] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018.

[4] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2020.

[5] J. Umlauft and S. Hirche, "Feedback linearization based on gaussian processes with event-triggered online learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4154–4169, 2019.

[6] J. B. Rawlings, D. Q. Mayne, M. Diehl *et al.*, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[7] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[8] 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 *2018 Annual American control conference (ACC)*. IEEE, 2018, pp. 1520–1527.

[9] X. Zhang, M. Bujarbaruah, and F. Borrelli, "Safe and near-optimal policy learning for model predictive control using primal-dual neural networks," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 354–359.

[10] J. A. Paulson and A. Mesbah, "Approximate closed-loop robust model predictive control with guaranteed stability and constraint satisfaction," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 719–724, 2020.

[11] M. Zanon and S. Gros, "Safe reinforcement learning using robust mpc," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2020.

[12] A. Mesbah, K. P. Wabersich, A. P. Schoellig, M. N. Zeilinger, S. Lucia, T. A. Badgwell, and J. A. Paulson, "Fusion of machine learning and MPC under uncertainty: What advances are on the horizon?" in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 342–357.

[13] R. Drummond, S. Duncan, M. Turner, P. Pauli, and F. Allgower, "Bounding the difference between model predictive control and neural networks," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 817–829.

[14] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. Springer Science & Business Media, 2013, vol. 31.

[15] A. Garulli, S. Paoletti, and A. Vicino, "A survey on switched and piecewise affine system identification," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 344–355, 2012.

[16] C. Mavridis and K. H. Johansson, "Real-time hybrid system identification with online deterministic annealing," *arXiv preprint arXiv:2408.01730*, 2024.

[17] C. N. Mavridis, A. Kanellopoulos, J. S. Baras, and K. H. Johansson, "State-space piece-wise affine system identification with online deterministic annealing," in *European Control Conference*, 2024, pp. 3110–3115.

[18] C. N. Mavridis and J. S. Baras, "Identification of piecewise affine systems with online deterministic annealing," in *IEEE Conference on Decision and Control*, 2023, pp. 4885–4890.

[19] C. Mavridis and J. S. Baras, "Annealing optimization for progressive learning with stochastic approximation," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 2862–2874, 2023.

[20] C. N. Mavridis and J. S. Baras, "Online deterministic annealing for classification and clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7125–7134, 2023.

[21] C. Mavridis and J. Baras, "Multi-resolution online deterministic annealing: A hierarchical and progressive learning architecture," *arXiv preprint arXiv:2212.08189*, 2022.

[22] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, no. Oct, pp. 1705–1749, 2005.

[23] C. N. Mavridis, N. Suriyarachchi, and J. S. Baras, "Maximum-entropy progressive state aggregation for reinforcement learning," in *IEEE Conference on Decision and Control*, 2021, pp. 5144–5149.

[24] C. N. Mavridis and J. S. Baras, "Progressive graph partitioning based on information diffusion," in *IEEE Conference on Decision and Control*, 2021, pp. 37–42.

[25] X. Lin, Z. Yang, X. Zhang, and Q. Zhang, "Continuation path learning for homotopy optimization," in *International Conference on Machine Learning*. PMLR, 2023, pp. 21 288–21 311.

[26] C. Mavridis, E. Noorani, and J. S. Baras, "Risk sensitivity and entropy regularization in prototype-based learning," in *Mediterranean Conference on Control and Automation*, 2022, pp. 194–199.

[27] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on mathematical software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.