Robust Data-Based Model Predictive Control for Nonlinear Constrained Systems ⋆

J.M. Manzano * **D. Limon** * **D. Muñoz de la Peña** * **J. Calliess** **

* *Department of Systems Engineering and Automation, Universidad de Sevilla, Spain (e-mails: {manzano,dlm,dmunoz}@us.es).*
** *Department of Engineering Science, University of Oxford, United Kingdom (e-mail: jan@robots.ox.ac.uk)*

Abstract: This paper presents stabilizing Model Predictive Controllers (MPC) to be applied to black-box systems subject to constraints in the inputs and the outputs. The prediction model of the controllers is inferred from experimental data of the inputs and outputs of the plant. Using a nonparametric machine learning technique called SPKI, the estimated (possibly nonlinear) model function is provided. Based on this, a predictive controller with stability guaranteed by design is proposed. Robust stability and recursive feasibility is ensured by using tightened constraints in the optimisation problem but without adding a terminal constraint on the optimisation problem. The proposed predictive controller has been validated in a simulation case study.

Keywords: MPC; Data-based control; Machine learning; Input-to-state stability.

1. INTRODUCTION

Model Predictive Control (MPC) is a controller design technique which uses a model of the plant to be controlled. If such model is unavailable a priori, it can be derived from observational data using machine learning methods.

The objective of this paper is to design a predictive controller which is robust and stable by design, based on these historian data. Hence, the learning method used must be adequate enough to estimate rich classes of dynamical systems and must be able to give bounds on the prediction error in order to ensure feasibility.

Following this approach, previous work on data-based MPC [Canale et al. (2014)] has utilised Nonlinear Set Membership (NSM) methods [Sukharev (1978); Milanese and Novara (2004)] for learning. These methods are capable of learning any Lipschitz continuous function with a known, given Lipschitz constant and provide bounds around the predictions of the learned model. [Canale et al. (2014)] presuppose a bound on the worst-case estimation error for the data set which they call the radius of information. Assuming that a stabilizing nominal MPC for the estimated prediction model has been designed, the authors prove that this controller is robustly stable. However, to do so, the authors not only need to assume that the bound on the Lipschitz constant and the radius of information are known, but they also assume that the nonlinear MPC optimisation problem happens to be recursively feasible, in spite of the prediction errors.

An alternative approach to data-based MPC that is more independent of the concrete learning paradigm was proposed in [Aswani et al. (2013)]. Here, the authors assumed that the true model was given by a linear model plus a function to be learned. The amplitude of the latter function had to be bounded and small, since it was considered to be an exogenous disturbance for the robust design of the linear robust MPC. The trained nonlinear model only entered the cost function while the nominal linear model was considered for the constraint satisfaction. Thus, the resulting feasibility region is convex, which allows the authors to derive robust stability provided that learning methods are employed that provide bounds around their predictions.

Since the uncertainty must enclose the estimation error and the mismatch between the linear and nonlinear dynamics, the resulting design might be quite conservative. Furthermore, the stability proof relies on the convexity of the constraints of the optimisation problem which cannot be extended to nonlinear models where convexity may not hold, requiring appropriate stability conditions.

In our paper, we improve on this existing state of affairs [Limon et al. (2017)] in several ways. Firstly, we propose to utilise nonlinear prediction models based on a recent improvement on NSM learning approach called SPKI which estimates Lipschitz constants from the data utilising a recently proposed hyperparameter optimisation approach [Calliess (2017)]. To improve applicability in meaningful MPC problems, our adapted approach smoothes the prediction signal and speeds up the computation by basing predictions on pre-partitioned subsets of the observed data. In order to enhance the estimation of the prediction model from data, the Lipschitz constant and the prediction is carried out component-wise.

On top of improvements to the learning and prediction method, we devise a nonlinear model-predictive controller in a manner that ensures robust stability and recursive feasibility. This is based on the calculation of bounds of the effect of the estimation errors along the predictions. The proposed bounds consider the component-wise nature of the estimation method and improves the method proposed by the authors in [Limon et al.

(2017)]. These bounds are used to tighten the constraints allowing us the satisfaction of the hard constraints on the output. Notably, the proposed controller is derived from an optimisation problem without terminal constraint while it ensures robust constraint satisfaction and input to state stability. This allowed us to avoid the cumbersome calculation of a suitable robust invariant set for the estimated prediction model, as proposed in [Limon et al. (2017)].

Notation and terminology

Given two column vectors a and b, (a, b) stands for $[a^T, b^T]^T$. Given two sets A, B, $A \oplus B$ represents the Minkowski sum, which is defined as the set $\{a + b : a \in A, b \in B\}$. $A \ominus B$ is the Pontryagin difference, defined as the set $\{c : c + b \in A, \forall b \in B\}$. The set $\mathbb{I}_a^b$ stands for the set of integers from a to b. A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a $\mathcal{K}$-function if it is strictly increasing and $\alpha(0) = 0$. If a $\mathcal{K}$-function is such that $\lim_{s \to \infty} \alpha(s) = \infty$ then it is called a $\mathcal{K}_\infty$-function. Remember, a pseudo-metric $\mathfrak{d}_\mathcal{W}(\cdot, \cdot) : \mathcal{W}^2 \to \mathbb{R}$ on a space $\mathcal{W}$ is a symmetric, positive mapping that adheres to the triangle-inequality. In contrast, to full metrics, a pseudo-metric does not require definiteness. That is, it is possible that $\mathfrak{d}_\mathcal{W}(w, w') = 0$ even if $w \neq w'$. A mapping $f : \mathcal{W} \to \mathcal{Y}$ between pseudo-metric spaces $(\mathcal{W}, \mathfrak{d}_\mathcal{W})$ and $(\mathcal{Y}, \mathfrak{d}_\mathcal{Y})$ is Hölder continuous with Hölder constant L and exponent p if $\mathfrak{d}_\mathcal{Y}(f(w), f(w')) \leq L \mathfrak{d}_\mathcal{W}(w, w')^p, \forall w, w' \in \mathcal{W}$. The smallest Hölder constant of f will be denoted by L^*.
Given a vector $v \in \mathbb{R}^{n_y}$ the ball $\mathcal{B}(v) \subset \mathbb{R}^{n_y}$ is defined as $\mathcal{B}(v) = \{y : |y_s| \leq v_s, s \in \mathbb{I}_1^{n_y}\}$. For a given compact set $A \subseteq \mathcal{W}$, $\mathfrak{d}_\mathcal{W}(A) = \max_{a \in A} \mathfrak{d}_\mathcal{W}(a, 0)$.

2. LEARNING THE PREDICTION MODEL

The prediction model of the controller used in this work will be inferred (i.e. learned) entirely from data, employing machine learning (i.e. regression) techniques. Specifically, we will use a class of learning rules which are referred to as *Kinky Inference* [Calliess (2014)]. It encompasses Lipschitz Interpolation [Sukharev (1978); Beliakov (2006)] and Nonlinear Set Interpolation [Milanese and Novara (2004)]. It's setting is as follows:

Let $\mathcal{W} \subset \mathbb{R}^{n_w}$, $\mathcal{Y} \subset \mathbb{R}^{n_y}$ be two spaces endowed with (pseudo-) metrics $\mathfrak{d}_\mathcal{W} : \mathcal{W}^2 \to \mathbb{R}_{\geq 0}$, $\mathfrak{d}_\mathcal{Y} : \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$, respectively. Spaces $\mathcal{W}, \mathcal{Y}$ will be referred to as *input space* and *output space*, respectively. It will be convenient to restrict our attention to input and output spaces that are additive abelian groups and which are *translation-invariant* with respect to their (pseudo) metrics. For simplicity, we will consider $\mathfrak{d}_\mathcal{Y}(y, y') = \|y - y'\| \forall y, y' \in \mathcal{Y}$.

Learning. It is our aim to learn a target function $f : \mathcal{W} \to \mathcal{Y}$ on the basis of a data set $\mathcal{D} = \{(s_i, \tilde{f}_i)|i = 1, ..., N_\mathcal{D}\}$ containing a sample $\mathcal{W}_\mathcal{D} := \{s_i | i = 1 \ldots, N_\mathcal{D}\}$ of inputs and noise-corrupted measurements of pertaining function values $\tilde{f}_i = f(s_i) + e(s_i)$ where $e(s_i)$ is some observational error $(i = 1, ..., N_\mathcal{D})$. Learning is the task of converting the data into a computable *predictor* $\hat{f} : \mathcal{W} \to \mathcal{Y}$ that allows us to infer *predictions* $\hat{f}(w)$ of $f(w)$ at unobserved *query inputs* $w \in \mathcal{W}$. In this work we will expand on the basis of a special case of the class of kinky inference predictors [Calliess (2014)] to perform learning as inference over unobserved function values. The prediction rule can be stated as follows:

Definition 1. (Kinky inference (KI) rule (simplified)). For $j = 1, ..., n_y$ the jth output component function of the KI predictor shall be defined by

$$\hat{f}_j(w; \theta, \mathcal{D}) := \frac{1}{2}\mathfrak{u}_j(w; \theta) + \frac{1}{2}\mathfrak{l}_j(w; \theta). \quad (1)$$

Here, $\mathfrak{u}_j(\cdot; \theta), \mathfrak{l}_j(\cdot; \theta) : \mathcal{W} \to \mathbb{R}^m$ are called ceiling and floor functions, respectively. There are many different ways of parameterising the metrics and defining the floor and ceiling functions (see e.g. Calliess (2017)). In this work, we assume $\theta = (L_1, ..., L_{n_y}, p_1, ..., p_{n_y})$; $\mathfrak{u}_j(w; \theta) := \min_{i=1,...,N_\mathcal{D}} \tilde{f}_{i,j} + L_j \mathfrak{d}^{p_j}(w, s_i)$ and $\mathfrak{l}_j(w; \theta) := \max_{i=1,...,N_\mathcal{D}} \tilde{f}_{i,j} - L_j \mathfrak{d}^{p_j}(w, s_i)$.

Note, given a fixed parameter θ, $\hat{f}_j$ is Hölder continuous with constant L_j and exponent p_j, relative to the pseudo-metric $\mathfrak{d}_\mathcal{W}$. The parameters therefore determine the smoothness and learning capacity properties of the predictor. While in several works, a priori knowledge of the correct parameters are presupposed [Zabinsky et al. (2003); Canale et al. (2014)], other works have proposed methods of adapting these parameters to the data [Milanese and Novara (2004); Calliess (2016, 2017)]. In this paper, we follow the so-called POKI method [Calliess (2017)] that optimises the parameters to minimise empirical risk and appears to be capable of smoothing out noise effectively without any distributional assumptions [1].

2.1 Smoothed Projected Kinky Inference (SPKI)

In this Section, we present our proposed prediction method, called SPKI. This method is based on two modifications of the KI method, namely the use of a reduced number of points to carry out the predictions, reducing the computation time; and the filtering of the outputs to smooth the result.

Predictor filtering The term *Kinky inference* makes reference to the result of the predictor described in Equation (1). Due to the kind of interpolation done, based on data and Hölder continuity of the function, the resulting prediction $\hat{f}$ is Hölder continuous but may be nondifferentiable, which may cause gradient optimisation methods to present solving issues. To avoid such problems we propose to use a version of the predictor known as *smoothed kinky inference* (SKI) [Calliess (2014)]. In this predictor, a convex combination of various points surrounding the query point w to be interpolated is used, as it is shown below:

$$f^*(w) := \sigma_0 \hat{f}(w) + \sum_{i=1}^{n_w} \frac{\sigma_i}{2}(\hat{f}(w + e_i \delta) + \hat{f}(w - e_i \delta)), \quad (2)$$

with weights $\sum_{i=0}^{n_w} \sigma_i = 1$, where e_i denotes the vector with a 1 in the i-th coordinate and 0's elsewhere, and δ is the incremental factor. It is easy to show that, by taking the convex combination, all Hölder properties of the predictor are preserved [Calliess (2014)].

Partitioning into local data sets In its standard form, the computation time to evaluate a prediction of the KI prediction (1) grows linearly with the number of training data points contained in the data set. For large data sets this can make

---

[1] For ease of notation, we will at time omit explicit mention of the dependence of the predictor on the data and parameters. For instance, we might write $\hat{f}(w)$ instead of $\hat{f}(w; \theta, \mathcal{D})$.

applicability to MPC intractable, as the prediction model will be evoked a large number of times during optimisation. To resolve this issue, we consider a partitioning and projection approach where predictions will be based only on a subset of the available data that allow us to limit the computational effort required to make a prediction. Our approach is motivated by the insight that for increasingly dense data sets, predictions of $f(q)$ for a query input $q \in \mathcal{W}$ should by and large only be affected by training examples $s \in \mathcal{W}_\mathcal{D}$ in close proximity to q. Bearing this idea in mind we propose the following approach: We partition the workspace into disjoint subsets and base the prediction of any given query point $q \in \mathcal{W}$ only on sample points contained in the same (and neighbouring) subsets. The input space $\mathcal{W} \subset \mathbb{R}^{n_w}$ of the data set $\mathcal{D}$ is partitioned into several subspaces $\mathcal{W}_1, ..., \mathcal{W}_m$. Next, for $i = 1, ..., m$ we define

$$\mathcal{D}_i = \{(w, \tilde{f}(w)) \in \mathcal{D} : w \in (\mathcal{W}_i \oplus \mathcal{B}(R_i))\}, \quad (3)$$

for certain overlapping radius R_i, to be the set of all samples whose inputs are contained in $\mathcal{W}_i$ or neighbouring subspaces.

This partition (and the subdata sets $\mathcal{D}_i$) can be calculated offline in a manner that can serve prespecified objectives such as ensuring that the expected number of samples per $\mathcal{W}_i$ does not exceed a threshold reflecting a given computational budget one desires to grant to the prediction algorithm.

Once these partitions have been computed, we can compute predictions of a query input $q \in W$ by firstly, determining the index i such that $q \in \mathcal{W}_i$ and secondly, by computing

$$\mathfrak{f}^\star(q; \theta, \mathcal{D}) := \mathfrak{f}^\star(q; \theta, \mathcal{D}_i). \quad (4)$$

The predictor $\mathfrak{f}^\star(\cdot)$ shall be referred to as *smoothed projected kinky inference (SPKI)*. Note, since computation of SPKI only relies on the subsample set $\mathcal{D}_i$, as long we have ensured that the number of samples per subsample set $\mathcal{D}_i$ is bounded (eg in expectation) by some threshold, the prediction time will be bounded accordingly. It can be proven that if the partitions data sets are chosen appropriately (that is, the value of R_i is large enough to capture all relevant data points), the prediction error can be equal to the prediction error of the inference obtained using the full data set.

3. DATA-BASED PREDICTION MODEL

The plant is composed of the inputs $u(k) \in \mathbb{R}^{n_u}$ and outputs $y(k) \in \mathbb{R}^{n_y}$, where $k \in \mathbb{N}$ stands for the sampling time. It is assumed that the system is described by the following *NARX* model of the plant

$$y(k+1) = g(x(k), u(k)) + e(k), \quad (5)$$

where $x(k) = (y(k), \cdots, y(k - n_a), u(k - 1), \cdots, u(k - n_b)) \in \mathcal{X} := \mathcal{Y}^{(n_a+1)} \times \mathcal{U}^{n_b} \subset R^{n_x}$ with $n_x = (n_a + 1)n_y + n_b n_u$, for some memory horizon lengths $n_a, n_b \in \mathbb{N}$. The signal $e(k)$ is the output noise and is assumed to be bounded. For notational convenience and relation with the previous section, we sometimes aggregate the inputs of g into a joint input $w := (x, u) \in \mathcal{W} := \mathcal{X} \times \mathcal{U}$. So, we can write $g : \mathcal{W} \to \mathcal{Y}$.

It is assumed that the origin is the equilibrium point of the system (i.e. $g(0, 0) = 0$) where the plant must be stabilized. Furthermore, the system is subject to constraints that must be satisfied at each sampling time:

$$u(k) \in \mathcal{U}, \qquad y(k) \in \mathcal{Y}, \quad (6)$$

where $\mathcal{U}$ and $\mathcal{Y}$ are compact sets.

Assumption 1. The setting is such that Assumption 1 from [Limon et al. (2017)] holds.

As in that paper, it is assumed that the prediction model will be inferred from a set of pairs of input-output experimental data

$$\mathcal{D} = \{(y(j), u(j)) \,|\, j = 1, \cdots, N_\mathcal{D}\}.$$

As defined in the previous section, the kinky inference method can be used to obtain a predictor $\hat{g}$ of the ground-truth g, yielding the output prediction:

$$\hat{y}(k+1) = \hat{g}(x(k), u(k); L_\mathcal{D}). \quad (7)$$

In order to give guarantees on the controller's closed-loop performance, we need to ensure recursive feasibility. Since our controller will be based not on the ground-truth dynamics g, but on the learned model $\hat{g}$, this can only be guaranteed if a bound on the discrepancy between g and $\hat{g}$ is known a priori and taken into account by the controller.

Assumption 2. It is assumed that a bound on the error between the estimated output and the real output is known. That is, there exists an a priori prediction error bound $\mu \in \mathbb{R}^{n_y}$ s.t.

$$|\hat{g}_i(x, u; L_\mathcal{D}) - g_i(x, u)| \leq \mu_i, \quad (8)$$

for all $i = 1, \ldots, n_y$, $x \in \mathcal{Y}^{n_a+1} \times \mathcal{U}^{n_b}$, and $u \in \mathcal{U}$.

4. STABILIZING DATA-BASED NMPC

In this section, a model predictive controller is designed for nonlinear systems subject to input and output constraints. This controller is proven to be robust and stable by design.

Based on the set of data $\mathcal{D}$ and the estimated Hölder constant $L_\mathcal{D}$, a prediction model $\hat{g}(x, u; L_\mathcal{D}, \mathcal{D})$ is obtained. For this analysis, it is convenient to define the prediction model of the plant in a state-space description [Limon et al. (2017)]. Then we can forecast the evolution of the system along a given prediction horizon N, for a given sequence of future inputs $u(k + j)$, $j = 0, \cdots, N - 1$ at a certain state $x(k)$ (5). Let $\hat{y}(j|k)$ denote the output that, at time k, is predicted to be observed at time $k + j$. Then the predicted state is given by

$$\hat{x}(j|k) = (\hat{y}(j|k), \cdots, \hat{y}(1|k), y(k), \cdots, y(k + j - n_a),$$
$$u(k + j - 1), \cdots, u(k + j - n_b))$$

where

$$\hat{x}(j + 1|k) = \hat{F}(\hat{x}(j|k), u(k + j)) \quad (9)$$

and

$$\hat{F}(\hat{x}(j|k), u(k + j)) = (\hat{g}(\hat{x}(j|k), u(k + j), L_\mathcal{D}),$$
$$\hat{y}(j|k), \cdots, y(k), \cdots,$$
$$y(k + j - n_a + 1), \cdots,$$
$$u(k + j), \cdots, u(k + j - n_b + 1)).$$

Lemma 1. Assume that at sampling time k, the state of the plant is $x(k)$ and a sequence of future control inputs $u(k + i)$ for $i \in \mathbb{I}_0^{N-1}$ is given.
Let $\hat{x}(j|k)$ and $\hat{y}(j|k)$ be the predicted states and outputs respectively derived from (9) for the given sequence of future control inputs and the current state $x(k)$, i.e. $\hat{x}(0|k) = x(k)$.
Assume that at sampling time $k + 1$, the current output $y(k + 1)$ is measured, and then the current state $x(k + 1)$ is known. Based on these new measurements, an updated sequence of states and outputs $\hat{x}(j|k + 1)$ and $\hat{y}(j|k + 1)$ are predicted based on (9)

with $\hat{x}(0|k+1) = x(k+1)$ and the remaining sequence of the given future control inputs.

Then the mismatch between both predictions satisfies

$$|\hat{y}_n(j-1|k+1) - \hat{y}_n(j|k)| \le c_{j,n}, \ n \in \mathbb{I}_1^{n_y}$$

where $c_j \in \mathbb{R}^{n_y}$ can be calculated from the recursion

$$c_{j+1,n} = L_{\mathcal{D},n}\,(r_j)^{p_n}$$

where $r_j = \mathfrak{d}_{\mathcal{X}}(\Xi_j)$, $c_1 = \mu$, $j \in \mathbb{I}_0^{N-1}$ and

$$\Xi_j = \mathcal{B}(c_j) \times \cdots \times \mathcal{B}(c_{\sigma(j)}) \times \underbrace{\{0\} \times \cdots \times \{0\}}_{n_b+1-\sigma(j-1) \text{ times}} \subseteq \mathbb{R}^{n_x}$$

with $\sigma(j) = \max(1, j - n_a)$ [2].

Remark 1. If the infinity norm is chosen as metric of the input and output space, then

$$r_j = \max_{s \in \mathbb{I}_{\sigma(j)}^j} \|c_j\|_\infty.$$

Based on the derived bounds on the prediction error, the problem of robust constraint satisfaction is analyzed next. To this aim, the following set of tightened constraints on the outputs is defined

$$\mathcal{Y}_j = \mathcal{Y} \ominus \mathcal{B}(d_j) \tag{10}$$

where

$$d_j = \sum_{s=1}^{j} c_s. \tag{11}$$

Lemma 2. The sets $\mathcal{Y}_j$ are such that for all $y \in \mathcal{Y}_j$ and for all $\Delta y \in \mathcal{B}(c_j)$, $y + \Delta y \in \mathcal{Y}_{j-1}$ [2].

In order to ensure that the proposed controller is feasible, the tightened set of constraints must be non-empty along the prediction horizon. This is stated in the following assumption:

Assumption 3. The prediction horizon N and the estimation error bound μ are such that the set $\mathcal{Y}_N$ is non empty.

Then the optimisation problem $P_N(x(k), L_{\mathcal{D}}, \mathcal{D})$ of the proposed predictive controller is the following

$$\min_{\mathbf{u}} V_N(x(k), \mathbf{u})$$

$$= \sum_{i=0}^{N-1} \ell(\hat{x}(i|k), u(i)) + \lambda V_f(\hat{x}(N|k)) \tag{12}$$

$$\text{s.t. } \hat{x}(0|k) = x(k) \tag{13}$$

$$\hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k), u(j)), \quad j \in \mathbb{I}_0^{N-1} \tag{14}$$

$$\hat{y}(j|k) = M\hat{x}(j|k) \tag{15}$$

$$u(j) \in \mathcal{U} \tag{16}$$

$$\hat{y}(j|k) \in \mathcal{Y}_j \tag{17}$$

where $M = [I_n, 0, \cdots, 0]$.

We require the ingredients of this optimisation problem to meet the following assumption:

Assumption 4.

(1) The stage cost function $\ell(x, u)$ is a Hölder continuous positive definite function such that $\ell(x, u) \ge \alpha_y(\mathfrak{d}_{\mathcal{X}}(x, 0))$ for a certain $\mathcal{K}$ function α and its Hölder constant is L_x.

[2] The proof of the theorems are omitted in this version due to the limited number of pages.

(2) There exists a control law $u = \kappa_f(x)$, a function V_f and a level set $\Omega_\gamma = \{x : V_f(x) \le \gamma\} \subseteq \mathbb{R}^{n_x}$ for a certain $\gamma > 0$ such that for all $x \in \Omega_\gamma$ the following conditions hold:

(a) V_f is a Hölder continuous positive definite function, with Hölder constant L_f, such that

$$\alpha_f(\mathfrak{d}_{\mathcal{X}}(x, 0)) \le V_f(x) \le \beta_f(\mathfrak{d}_{\mathcal{X}}(x, 0))$$

$$V_f(\hat{F}(x, \kappa_f(x))) - V_f(x) \le -\ell(x, \kappa_f(x)).$$

(b) $\kappa_f(x) \in \mathcal{U}$, $Mx \in \mathcal{Y}_N$.

Define the function

$$\upsilon(\mu) = \sum_{j=1}^{N} L_x r_j(\mu)^p + \lambda L_f r_{N+1}(\mu)^p,$$

where r_j is defined in Lemma 1.

Assumption 5.
It is assumed that μ is such that the set

$$\Upsilon = \{x : \ell(x, 0) \le \upsilon(\mu)\}$$

is contained in Ω_γ. It is also assumed that the positive constants λ and ϕ are such that $\lambda \ge 1$ and $\ell(x, 0) > \phi$ for all $x \notin \Omega_\gamma$.

Lemma 3. Under the Assumption 5, the constant ϕ is such that $\phi \ge \upsilon(\mu)$ [2].

Let Γ define the following level set of the optimal cost function

$$\Gamma = \{x : V_N^*(x) \le N\phi + \lambda\gamma\}.$$

Notice that this set is compact and non-empty.

Theorem 1. (ISS stability). Suppose that Assumptions 2, 3, 4 and 5 hold for the optimisation problem $P_N(\cdot)$. Let $\kappa_N(x)$ be the control law derived from the solution of $P_N(x)$ applied using a receding horizon policy. Then, for any $x(0) \in \Gamma$, the system controlled by the control law $u(k) = \kappa_N(x(k))$ is input-to-state stable w.r.t the estimation error and the constraints are always satisfied, i.e. $y(k) \in \mathcal{Y}$ and $x(k) \in \Gamma, \forall k$ [2].

5. APPLICATION

5.1 The continuously-stirred tank reactor

In this Section we consider the control of a continuously stirred-tank reactor presented in [Seborg et al. (1989)]. The experiments carried out to generate the data sets and the procedure to tune the proposed controller are described in what is to follow.

The input of the reactor is the reference of the coolant temperature T_r (K) and the outputs are the concentration of the reactant, C_A (mol/l), in the reaction $A \rightarrow B$, the temperature of the tank and the temperature of the coolant, T and T_c respectively (K).

It is assumed that the evolution of the plant is given by the following set of differential equations:

$$\frac{dC_A(t)}{dt} = \frac{q_0}{V} \cdot (C_{Af} - C_A(t)) - k_0 \cdot exp\left(-\frac{E}{R \cdot T(t)}\right) \cdot C_A(t) \tag{18}$$

$$\frac{dT(t)}{dt} = \frac{q_0}{V} \cdot (T_f - T(t)) +$$

$$+ \frac{(-\Delta H_r) \cdot k_0}{\rho \cdot C_p} \cdot exp\left(-\frac{E}{R \cdot T(t)}\right) \cdot C_A(t) +$$

$$+ \frac{U \cdot A}{V \cdot \rho \cdot C_p} \cdot (T_c(t) - T(t)) \tag{19}$$

$$\frac{dT_c(t)}{dt} = \frac{T_c(t) - T_r(t)}{\tau}. \tag{20}$$

The parameters of the model are given in Table 1. Note that the model is only used to carry out simulations, no information is used to design the controller. It is also assumed that the three output sensors adds a measurement error of 2%. The error is generated randomly for each measurement using an uniform distribution. The constraints in the input are $335\,\text{K} \leq T_r \leq 372\,\text{K}$, and the constraints in the outputs are given by $0.1982 \leq C_A \leq 0.8014(\text{mol/l})$, $333.538 \leq T \leq 376.845(\text{K})$ and $331.748 \leq T_c \leq 375.6775(\text{K})$.

Table 1. Parameters of the system

Param.	Definition	Value	Units
q_0	Input flow of the reactive	10	l/min
V	Liquid volume in the tank	150	l
k_0	Frequency constant	6×10^{10}	1/min
E/R	Arrhenius constant	9750	K
$-\Delta H_r$	Enthalpy of the reaction	10000	J/mol
UA	Heat transfer coefficient	70000	J/(min K)
ρ	Density	1100	g/l
C_p	Specific heat	0.3	J/(g K)
τ	Time constant	1.5	min
C_{Af}	C_A in the input flow	1	mol/l
T_f	Temperature (input flow)	370	K

5.2 Obtaining the data set

The sampling time was set to $30\,\text{s}$. An equilibrium point was chosen to be the reference operating point, i.e. $C_A^{\text{ref}} = 0.439\,\text{mol/l}$, $T^{\text{ref}} = 359.75\,\text{K}$, $T_c^{\text{ref}} = 356.68\,\text{K}$ and $T_r^{\text{ref}} = 356\,\text{K}$. A set of experiments were carried out to obtain the data for the predictor. The experiments were designed using the methodologies presented in [Rivera and Jun (2000)]: a sequence of chirp signals covering the workspace were applied to generate the raw data set containing the trajectories of concentrations and temperatures, $\mathcal{D}_{\text{raw}}$ [3]. The resulting data points are represented in Figure 1.

In addition, several tests with random input signals were carried out in order to obtain data sets for cross-validation, $\mathcal{D}^{\text{val}}$.

The data set $\mathcal{D}_{\text{raw}}$ was scaled, obtaining the scaled input and output trajectories data set $\mathcal{D}_{\text{exp}} = \{y, u\}$ using the following expressions (also used for the maximum, minimum and reference values):

$$u = \frac{T_r - T_r^{\min}}{T_r^{\max} - T_r^{\min}}$$

$$y = \left[\frac{C_A - C_A^{\min}}{C_A^{\max} - C_A^{\min}}, \frac{T - T^{\min}}{T^{\max} - T^{\min}}, \frac{T_c - T_c^{\min}}{T_c^{\max} - T_c^{\min}} \right]. \tag{21}$$

To obtain the SKI model, the parameters n_a and n_b of the NARX regressor had to be chosen. Since the outputs coincided with the canonical realization of the system, we chose $n_a = n_b = 0$, that is, $x_k = y_k$. The data set $\mathcal{D}^{\text{cond}} = \{(x_k, u_k), y_{k+1}\}$, with $k = 1 \dots N_{\mathcal{D}}$, was constructed from the experimental data as explained in Section 3. The SPKI predictor can be defined as in (4). The SKI filtering of Equation (2) was done predicting with $\sigma_0 = {}^2/(n_x + 2)$, $\sigma_i = {}^1/n_x + 2$ and increment δ set to 0.015.

[3] For more details on the design parameters and results of the case study, please get in touch with the corresponding author.
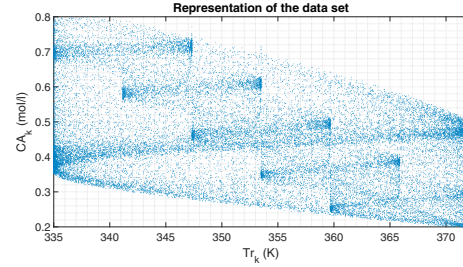

Fig. 1. 2-D representation of $\mathcal{D}^{\text{cond}}$. Pair $\{C_{A_k}, T_{r_k}\}$ for k=1,...,30000.

To this end, the Hölder constants $L_{\mathcal{D}}$ and exponents p were obtained using a modification of the *POKI* [Calliess (2017)]. Instead of minimizing the empirical one-step-ahead prediction error μ, we set the parameters to minimize the maximum error within the prediction horizon, i.e., d_j (11).

Setting $N = 4$ and given $\mathcal{D}^{\text{cond}}$ and $\mathcal{D}^{\text{val}}$, for $L_{\mathcal{D}} = [0.841\ 0.772\ 0.745]^4$ and $p = [0.8815\ 0.898\ 0.883]^4$ the maximum propagation of the prediction error was minimized. For this configuration $\mu = [0.020\ 0.159\ 0.0865]^4$.

In order to reduce the computation effort, the regressor space $\mathcal{W}$ was divided into a set of overlapping regions following the procedure outlined in Section 2.1. The prediction for 100 random query points in an Intel® Core™ i7-6700HQ CPU @ 2.60GHz 12GB RAM without the partitions division took $41.826\,\text{s}$, while using partition-based approach took only $0.0717\,\text{s}$, without worsening the prediction error.

5.3 Control of the reactor

The optimisation problem (12) was solved in MATLAB using the optimisation function *fmincon*. The reference was set in the equilibrium point of the system given by $\{y^{\text{ref}}, u^{\text{ref}}\}$, defined in the previous subsection.

The stage cost was defined as a quadratic cost ℓ:

$$\ell(x - x_r, u - u_r) = ||x - x_r||_Q^2 + ||u - u_r||_R^2$$

The parameters were set to $Q = \text{diag}[100\ 10\ 10]$ and $R = 1$. As terminal ingredients, $\kappa_f(x) = K(x - x_r) + u_r$ and $V_f(x) = ||x - x_r||_P^2$ were considered. This was calculated from the solution of the LQR for a linear model around the reference. This linearized model, calculated numerically from the input-output data, had the form $x_{k+1} = Ax_k + Bu_k$, where

$$A = \begin{bmatrix} 0.955 & 0.004 & 0.129 \\ -0.159 & 0.4525 & 0.343 \\ 0.0634 & 0.161 & 0.1845 \end{bmatrix} \quad B = \begin{bmatrix} -0.370 \\ 0.109 \\ 0.357 \end{bmatrix}.$$

To ensure robust stability (Theorem 1), Assumptions 2, 3, 4 and 5 must hold true. μ was obtained via cross-validation. With the values stated above, d_N (Equation 11) was such that $\mathcal{Y}_N = \{y : [0.102\ 0.115\ 0.102] \leq y \leq [0.898\ 0.885\ 0.898]\}^4$, so it is non-empty. Finally, following the procedure in Section 4 and in [Limon (2002)] we obtained $\gamma = 5.356$, $\upsilon = 56.099$ and $\phi = 152.98$; and we checked that $\Upsilon \subset \Omega_\gamma$.

Once the controller was designed, a hundred simulations were carried out, with the same initial state but a hundred different realizations of the random noise. Figure 2 shows the SPKI-MPC, based entirely on historical data of inputs and outputs
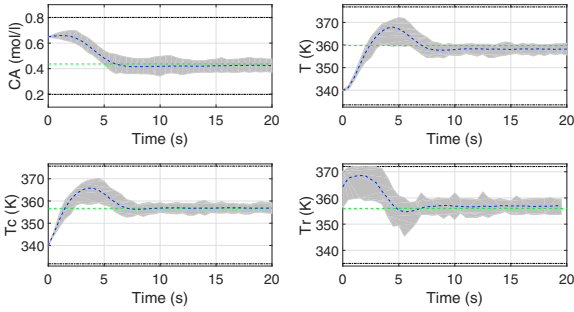
[4] Note that the values were scaled as in (21).

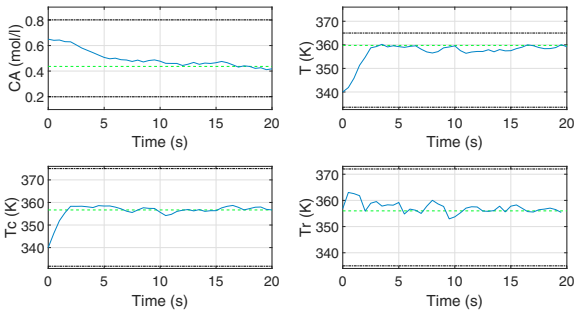Fig. 2. Data-based predictive control applied to a CSTR.

Fig. 3. Simulation with smaller upper constraint in T.

of the plant. The Figure represents the three outputs and the control input. Within each subplot, the constraints appear in black, the reference in green, and the 100 trajectories are represented by the gray *band*, whose mean value is plotted in blue. Figure 3 shows a test in which the upper constraint on T is decreased to 365 K.

Apart from that, another hundred different simulations of the MPC (with random initial state) were carried out, which results are shown in Table 2. This comparison includes the best closed-loop behavior achievable assuming perfect knowledge of the model (its set of ODEs) and the unmodified KI method, that is, the MPC in which neither the smoothing nor the partitions were added to the prediction model.

The performance index of each simulation is defined as:

$$\Phi = \sum_{i=1}^{t_{\text{sim}}} \ell(x(i), u(i)). \tag{22}$$

Table 2. Performance of the MPCs

MPC	ODEs	Primary KI	SPKI
Mean performance cost Φ	122.37	171.76	145.32
Standard deviation	13.05	24.21	17.742
Mean time per iteration (s)	0.27	19.12	0.86

6. CONCLUSIONS

A MPC control approach whose model is based on data was proposed. This model is learned using an improved version of the inference method called kinky inference. Given input-output observations of the plant, the MPC is able to solve an optimisation problem to obtain the sequence of control inputs to be applied. Input-to-state stability and recursive feasibility

of the closed-loop system are proven under some assumptions. To illustrate the viability of our ideas, simulations of a continuously-stirred tank reactor were provided. Here the learning-based MPC proven capable of learning how to drive the signals and stabilize in the presence of noise.

In addition, we derived tighter enhanced bounds on the effect of the multiple step-lookahead uncertainty. And, based on these, we proposed a novel predictive controller capable to ensure robust stability without terminal constraints. In future work, we will investigate the utilisation of other inference methods such as Gaussian processes within the learning-based MPC. We will also consider online learning as well as techniques to reduce the conservatism of the proposed approaches.

REFERENCES

Aswani, A., Gonzalez, H., Sastry, S.S., and Tomlin, C. (2013). Provably safe and robust learning-based model predictive control. *Automatica*, 49(5), 1216–1226.

Beliakov, G. (2006). Interpolation of Lipschitz functions. *Journal of Computational and Applied Mathematics*.

Calliess, J.P. (2014). *Conservative decision-making and inference in uncertain dynamical systems*. Ph.D. thesis, University of Oxford.

Calliess, J.P. (2016). Lazily adapted constant kinky inference for regression and nonparametric model-reference adaptive control. *ArXiv pre-print, arXiv:1701.00178*.

Calliess, J.P. (2017). Lipschitz optimisation for Lipschitz interpolation. In *American Control Conference (ACC), 2017*, 3141–3146. IEEE.

Canale, M., Fagiano, L., and Signorile, M.C. (2014). Nonlinear model predictive control from data: a set membership approach. *International Journal of Robust and Nonlinear Control*, 24(1), 123–139.

Limon, D. (2002). *Control Predictivo de sistemas no lineales con restricciones: estabilidad y robustez*. Ph.D. thesis, Universidad de Sevilla.

Limon, D., Calliess, J., and Maciejowski, J. (2017). Learning-based nonlinear model predictive control. In *Proceedings of the 20th IFAC World Congress*.

Milanese, M. and Novara, C. (2004). Set membership identification of nonlinear systems. *Automatica*.

Rivera, D.E. and Jun, K.S. (2000). An integrated identification and control design methodology for multivariable process system applications. *IEEE Control Systems*, 20(3), 25–37.

Seborg, D.E., Edgar, T.F., and Mellichamp, D.A. (1989). *Process Dynamics and Control*. Wiley.

Sukharev, A. (1978). Optimal method of constructing best uniform approximation for functions of a certain class. *Comput. Math. and Math. Phys.*

Zabinsky, Z.B., Smith, R.L., and Kristinsdottir, B.P. (2003). Optimal estimation of univariate black-box Lipschitz functions with upper and lower bounds. *Computers and Operations Research*.