

Output Feedback MPC based on Smoothed Projected Kinky Inference

J.M. Manzano¹, D. Limon¹, D. Muñoz de la Peña¹, J.P. Calliess²

Abstract—In this paper we propose a stabilizing data-based model predictive controller for systems subject to constraints in which the prediction model is inferred from experimental data of the plant using a machine learning technique. The inference method is a modification of the kinky inference tailored for model predictive control. In particular, the modified method has a lower computational effort and provides smoother predictions than the original method. The controller formulation considers soft constraints in the outputs, hard constraints in the inputs and guarantees closed-loop robust stability as well as performance by means of the use of different control and prediction horizons and a weighted terminal cost. Under the assumption that the model of the system is Hölder continuous, we prove that the closed-loop system is input-to-state stable with respect to the estimation errors. The results are demonstrated in a case study of a continuously stirred-tank reactor.

I. INTRODUCTION

Model predictive control (MPC) [1] is based on using a dynamical model of the plant to predict the behavior of a system given some sequence of future control actions. The predictions are used to estimate the cost associated to the predicted trajectory as well as the degree of satisfaction of the constraints.

Since the decisions on control actions rest on the predictive model, its accuracy is key. If accurate predictive models can be derived from first principles, a wealth of work is applicable that allow the design of MPC controllers with strong theoretical performance guarantees. Unfortunately, in reality, such models may not always be available. This may be due to the complexity of their derivation or owing to the controller operating in changing environments that exhibit a priori unforeseen dynamics.

For such scenarios, data- or learning-based MPC methods have attracted increasing attention in recent years. These methods have the capability to infer prediction models based on historically available data. Traditionally, this most often has been done offline by means of dynamic systems identification techniques, often suitable to approximate the dynamics of the system around a given operating point using a linear model [2].

Recently, different predictive control schemes which are based on non-linear machine learning methods have been proposed in the literature. For example, Aswani et. al. [3] have considered modified Nadaryana-Watson estimators to estimate the global uncertainty of the system, in order to improve predictions of an a priori linear model. Subsequently,

Canale et. al. [4] proposed a data-based nonlinear predictive controller that implicitly uses Lipschitz interpolation techniques based on the assumption that a Lipschitz constant of the dynamics as well as a finite bound on the noise are known a priori. Unfortunately, it remained unclear how these parameters can be chosen in practice and what the impact of choosing them incorrectly might be. Thereby, it remains an open question in how far this approach does exhibit sufficient black-box learning capabilities to be considered a fully fledged data-based controller.

In this paper we propose an extension and generalization of this setup. In contrast to earlier works based on Lipschitz interpolation, our approach is based on the more general class of kinky inference methods [5]. Furthermore, the parameters corresponding to the Lipschitz constants are also estimated from the data, following the procedure in [6], known as parameter optimised kinky inference (POKI).

As in standard nonparametric regression methods, the computational effort when using POKI for making predictions grows with the sample size. Since in MPC, the prediction model is frequently evoked to compute large numbers of predictions, reducing computational effort for prediction is important. To this end, we propose a variation of kinky inference that is designed to speed up predictions. Instead of making predictions on the basis of all observed data, our approach is based on using a different reduced subset of data for each region of a partition of the state space to implement the smooth kinky inference method presented in [5]. The use of different reduced data sets depending on the current state decreases the computational effort of the inference method. The smooth predictions yield smooth control actions that result in better performance. The whole method is called smooth projected kinky inference (SPKI), which is an extension of existing data-based controllers such as [7], [8].

In addition, a new controller formulation that guarantees closed-loop robust stability as well as performance by means of the use of different control and prediction horizons and a weighted terminal cost is proposed. This controller is an extension of the controller proposed in [9], which provides a stabilizing design without the use of a terminal constraint. The new controller takes into account an output feedback formulation. Under the assumption that the model of the system is Hölder continuous, we prove that the closed-loop system is input-to-state stable with respect to the measurements noises and prediction errors.

The proposed strategy can be designed using input-output data collected from the application of standard test signals and identification procedures [2] and, because of the gen-

¹Department of Systems Engineering and Automation, Universidad de Sevilla, Spain. {manzano, dlm, dmunoz}@us.es

² Department of Engineering Science, University of Oxford, United Kingdom. jan-peter.calliess@oxford-man.ox.ac.uk

erality of its formulation (output feedback control of input constrained systems), it can be applied to a wide set of control problems. The proposed controller has been validated in simulation using the control of a continuous stirred tank reactor case study.

The rest of the paper is structured as follows: Section 2 states the dynamical scheme of the control, defining the scope of the problem. Section 3 introduces the prediction scheme, which includes the definition of the prediction function, the design of its parameter and a couple of modification to enhance the performance of the method. Section 4 introduces the model predictive controller and its ingredients to ensure robust stability and recursive feasibility. Finally, Section 5 illustrates the settings in a case study: a continuously-stirred tank reactor.

Notation

Given two column vectors a, b , the notation (a, b) implies $[a^T, b^T]^T$. Given a set $\mathcal{Y} \subset \mathbb{R}^p$ and a point $y \in \mathbb{R}^p$, the distance of the point to the set is defined as $\mathfrak{d}(y, \mathcal{Y}) = \min_{z \in \mathcal{Y}} \|z - y\|_\infty$ where $\|\cdot\|$ stands for a norm in the euclidean space and $\|\cdot\|_\infty$ denotes the infinity norm. The ball of radius R is defined as $x \in \mathcal{B}(R) = \{x : \|x\|_\infty \leq R\}$. The set \mathbb{I}_a^b stands for set of integers from a to b . $|A|$ implies the cardinality of the set A . Given two sets A, B , the Minkowski sum $A \oplus B$ is defined as the set $\{a + b : a \in A, b \in B\}$. Given a set A and a space B , $\text{Proj}_B(A)$ stands for the projection of A in B . A function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a \mathcal{K} function if $\alpha(0) = 0$ and it is strictly increasing. It is a \mathcal{K}_∞ function if besides that, it is unbounded. A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a \mathcal{KL} function if $\beta(s, t)$ is \mathcal{K}_∞ in s for any value of t and $\lim_{t \rightarrow \infty} \beta(s, t) = 0$ for all $s \geq 0$.

II. CONTROL OBJECTIVE

The objective of this research is to develop robustly stabilizing output feedback predictive controllers whose model is entirely based on data, for a nonlinear system subject to hard constraints in the manipulable inputs $u(k) \in \mathcal{U} \subset \mathbb{R}^{n_u}$ and soft constraints in the measured controlled outputs $y(k) \in \mathcal{Y} \subset \mathbb{R}^{n_y}$ such that it is steered to a given reachable reference y_r . It is assumed that the reference is an equilibrium point that satisfies the constraints, such that $y_r \in \mathcal{Y}$ and $u_r \in \mathcal{U}$.

It is considered that the system can be described by a NARX model [10], i.e. the dynamics can be written as follows:

$$y(k+1) = f(x(k), u(k)) \quad (1)$$

where the state is

$$x(k) = (y(k), \dots, y(k-n_a), u(k-1), \dots, u(k-n_b)) \in \mathbb{R}^{n_x} \quad (2)$$

with $n_x = (n_a + 1)n_y + n_b n_u$, for some memory horizons $n_a, n_b \in \mathbb{Z}$. Note that $x(k)$ is defined for $n_b \geq 1$; however, we could have $n_b = 0$, to indicate that the state x_k only depends on the past outputs.

It is considered that the model function f is unknown. Instead, in order to infer the future evolution of the system, a set of input and output trajectories obtained from experimental data is available. The data set is denoted by \mathcal{D} . To this end, a machine learning inference method that uses directly the information in the data set \mathcal{D} to predict the system model will be used. This prediction model is nonexplicit and it is based on a machine learning method proposed in [6] known as ‘‘parameter optimised kinky inference’’, POKI.

The proposed controller decides the value of the current inputs, given the latest measurements and inputs, the reference and the data set:

$$u(k) = \kappa_{\text{MPC}}(x(k), y_r; \mathcal{D})$$

III. TRAJECTORY PREDICTION WITH SMOOTHED PROJECTED KINKY INFERENCE

A. Kinky inference

The kinky inference (KI) approach is a class of nonparametric regression methods that are described in [5], [11]. For completeness, a brief outline of a subclass of these methods is provided in this Section. The method is used to learn a map $f : \mathcal{W} \rightarrow \mathcal{Z}$. The map is assumed to be Hölder continuous with Hölder constant L^* and exponent $0 < \alpha \leq 1$, so $\forall w_1, w_2 \in \mathcal{W}$ the following condition holds:

$$\|f(w_1) - f(w_2)\| \leq L^* \|w_1 - w_2\|^\alpha$$

From this map f , certain pairs of inputs and outputs (possibly noisy) are known, forming the data set

$$\mathcal{D} : = \{(w_i, \tilde{f}(w_i)) | i = 1, \dots, N_{\mathcal{D}}\}$$

where \tilde{f} stands for the noisy observation of f . The set containing only the input data points is denoted as $\mathcal{W}_{\mathcal{D}} = \text{Proj}_{\mathcal{W}}(\mathcal{D})$. Although the true function f remains unknown, we assume that a bound on the observational error ($\bar{\epsilon}$) is known, so $\forall w \in \mathcal{W} : \|\tilde{f}(w) - f(w)\| \leq \bar{\epsilon} \in \mathbb{R}_{\geq 0}$.

Remark 1: Note that if $\alpha = 1$ the Hölder definition becomes the so-called Lipschitz continuity. However, the extension to Hölder allows one to handle more general systems where the Lipschitz continuity does not hold true, whereas the Hölder continuity does. For instance, consider the case of a tank with two holes, one at the bottom and the other one at $H = 1$ m. If the water height were to be controlled in a compact set such that $0.8 \text{ m} \leq h \leq 1.2 \text{ m}$ (e.g.), the NARX model of the problem is:

$$h(k+1) = h(k) + \frac{T_m}{A} (q_{in}(k) - q_{out}(h(k)))$$

where T_m is the sampling time, A is the area of the tank, q_{in} the input flow and $q_{out} = k_1 \sqrt{h} + k_2 \sqrt{\max\{(h - H), 0\}}$ the output flow.

Figure 1 shows the function output flow q_{out} to be identified for $k_1 = k_2 = 1$. The Lipschitz constant $L^*(\alpha = 1)$ becomes infinite for $h=1$ m, while the Hölder continuity still holds for $\alpha < 1$.

Given an input $q \notin \mathcal{W}_{\mathcal{D}}$, the estimated value of the map \tilde{f} is built in the following way [5]:

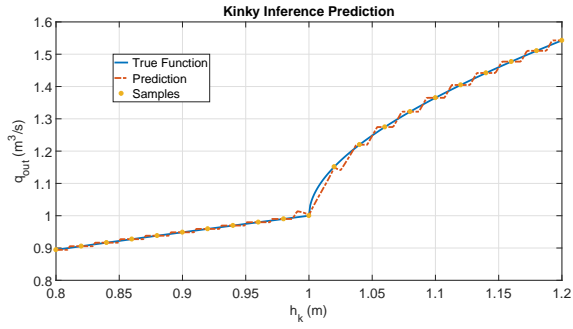


Fig. 1. Demonstration of the operation of KI with 20 samples in h , $L=1.25$ and $\alpha = 0.8$ for q_{out} .

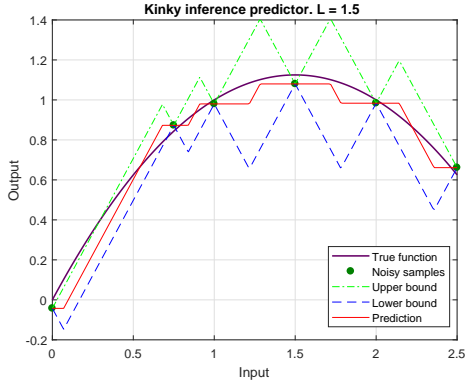


Fig. 2. Kinky Inference for $f(x) = -\frac{x^2}{2} + \frac{3x}{2}$.

$$\hat{f}_j(q; \theta, \mathcal{D}) = \frac{1}{2} \min_{i=1, \dots, N_{\mathcal{D}}} (\tilde{f}_{i,j} + L_{\mathcal{D}} \|q - w_i\|^{\alpha}) + \frac{1}{2} \max_{i=1, \dots, N_{\mathcal{D}}} (\tilde{f}_{i,j} - L_{\mathcal{D}} \|q - w_i\|^{\alpha}) \quad (3)$$

where $\hat{f}_j(q; \theta, \mathcal{D})$ is the j -th component of \hat{f} , $\tilde{f}_{i,j}$ is the j -th component of the value of the observed map for the i -th data point in \mathcal{D} , w_i is its corresponding input, $N_{\mathcal{D}}$ is the number of data points and θ are the parameters $\{L_{\mathcal{D}}, \alpha\}$.

Figure 1 also shows the prediction for $N_{\mathcal{D}} = 20$ samples, $L_{\mathcal{D}} = 1.25$ and $\alpha = 0.8$. A simpler example of the geometric interpretation of the Kinky Inference method is shown in Figure 2, where $f(x) = -\frac{x^2}{2} + \frac{3x}{2}$ is predicted based on 6 noisy samples, with $L = 1.5$ and $\alpha = 1$.

The value of the Hölder constant and the exponent of the function are unknown. While in several works, a priori knowledge of the correct Hölder constant is presupposed [12], [13], other works have proposed methods of adapting these parameters to the data [14], [11], [6]. In this paper, we follow the so-called POKI method [6], which optimises both parameters to minimize empirical risk and appears to be capable of smoothing out noise effectively without any distributional assumptions.

To this end, two different data sets are used: one for conditioning and predicting, $\mathcal{D}^{\text{cond}}$, and another one for validation, \mathcal{D}^{val} , where the prediction error is to be minimized. Hence, grouping $\theta = (L_{\mathcal{D}}, \alpha)$,

$$\theta := \arg \min_{\theta} \frac{1}{|\mathcal{W}^{\text{val}}|} \sum_{w \in \mathcal{W}^{\text{val}}} \|\tilde{f}(w) - \hat{f}(w; \theta, \mathcal{D})\| \quad (4)$$

where $\mathcal{W}^{\text{val}} = \text{Proj}_{\mathcal{W}}(\mathcal{D}^{\text{val}})$.

It is important to remark that the maximum estimation error in the KI predictor, defined as $\mu = \sup \|\tilde{f}(w) - \hat{f}(w)\|$, decreases tending to $\bar{\epsilon}$ when the density of the input data set increases, being able to predict the observed map $\tilde{f}(\cdot)$ with arbitrary small error from a data set dense enough, provided that the real map is Hölder continuous.

The kinky inference method has multiple advantages, some of which are:

- Learning simplicity. It requires only two hyperparameters, the Hölder constant $L_{\mathcal{D}}$ and exponent α . Obtaining the unknown constant given a noisy data set is also easy to compute [6], [11].
- Online configurability. It is very simple to include new data points, updating the predictor.
- Parallel prediction. The prediction is calculated by a loop over the data points; hence, it is easily parallelizable.
- Numerically stable, as opposed to Gaussian processes; and computationally simpler - it can be computed in few cycles on micro-controllers.

B. Prediction model

Given an experimental data set $\mathcal{D}_{\text{exp}} = \{(y_k, u_k), \forall k \in \mathbb{I}_1^{N_{\text{exp}}}\}$ including a sequence of input and output trajectories, and a fixed n_a and n_b , the data set for the prediction model can be constructed as

$$\mathcal{D} := \{(w_k, z_k) | k = 1, \dots, N_{\mathcal{D}}\}$$

defining $w_k = (x(k), u(k)) \in \mathbb{R}^{n_w}$ with $n_w = n_x + n_u$ and $z_k = y(k+1) \in \mathbb{R}^{n_y}$. The prediction model can be calculated from \mathcal{D} by means of the KI method and it can be written as $z_k = \hat{f}(w_k; \theta, \mathcal{D})$. With a slight abuse of notation, this is expressed as follows:

$$\hat{y}(k+1) = \hat{f}(x(k), u(k); \theta, \mathcal{D}) \quad (5)$$

Remark 2: Sometimes prior knowledge of the system is available. Consider the system

$$y(k+1) = g(x(k), u(k)) + f(x(k), u(k)), \quad (6)$$

where $g(x, u)$ stands for the known dynamics (e.g. a linear behavior), and $f(x, u)$ represents some unmodeled dynamics. In this favorable case, it is always possible to include this prior knowledge of the system in the prediction of the evolution, and to learn and to predict, based on data, the unmodeled part:

$$\hat{y}(k+1) - g(x(k), u(k)) = \hat{f}(x(k), u(k); \theta, \mathcal{D}).$$

As motivated in the previous Section, the KI method enjoys properties that make it suitable for the estimation of prediction models. However, the computational effort depends linearly on the size of the data set. Furthermore, the

prediction may not be smooth, which may limit its real time applications. This motivates the study of methods to reduce the computation cost and to smooth the predictions. This is particularly relevant in the derivation of prediction models for model predictive control, because, as it was mentioned before, the control techniques require the solution of an optimisation problem at each sampling time.

C. Smoothed projected kinky inference

In this Section we present the proposed SPKI method. 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.

1) *Smooth Kinky Inference*: The term kinky inference makes reference to the result of the predictor described in equation (3). 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 (see Figure 1), 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) [5]. In this predictor, a convex combination of various points surrounding the query point w to be interpolated is used, as it is shown below:

$$f^*(w; \theta, \mathcal{D}) = \sigma_0 \hat{f}(w) + \sum_{i=1}^{n_w} \frac{\sigma_i}{2} (\hat{f}(w + e_i \delta) + \hat{f}(w - e_i \delta)) \quad (7)$$

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. Note that the dependence of \hat{f} w.r.t. to θ and \mathcal{D} is omitted in favor of conciseness. The new predictor f^* is Hölder continuous and its Hölder constant is the same as in the prediction function $\hat{f}(\cdot)$ [5].

The KI predictions at each one of the dimensional components of w can be used to calculate an approximation of the gradient of f^* , and by extension, to estimate the gradient of the cost. This gradient can be given to the optimiser in order to increase the iteration speed. Such gradient is approximated by:

$$\frac{\partial f^*}{\partial w_i} \simeq \frac{\hat{f}(w + e_i \delta) - \hat{f}(w - e_i \delta)}{2\delta}$$

2) *Projected Kinky Inference*: In its standard form, the computation time to evaluate a prediction of the KI prediction grows linearly with the number of training data points contained in the data set. As predictions occupy a large amount of time during the repeated optimisations performed by our MPC controller, this property constitutes a serious computational bottleneck. In this Section, we will address this issue. To this end, we partition the input workspace \mathcal{W} 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 neighboring) subsets.

The input space $\mathcal{W} \subset \mathbb{R}^{n_w}$ of the data set \mathcal{D} is divided into several partitions \mathcal{W}_i , such that the union of them conform

the original space $\mathcal{W} = \cup(\mathcal{W}_i)$, and their intersection is null: $\cap_{i \neq j} \mathcal{W}_i = \emptyset, \forall j$. This partition is calculated offline and then a classification algorithm is built to locate the partition \mathcal{W}_i to which a query point belongs to. This partition can be calculated taking into account different objectives, as for instance, to ensure a regular distribution of data points.

Once the set of the partition where the query $q \in \mathcal{W}$ is located is found, the prediction is computed as

$$f^*(q; \theta, \mathcal{D}) := f^*(q; \theta, \mathcal{D}_i), \quad (8)$$

only taking into account sample points of a local subset of data points \mathcal{D}_i where

$$\mathcal{D}_i = \{\mathcal{D} | w \in \mathcal{W}_D : w \in (\mathcal{W}_i \oplus \mathcal{B}(R_i))\} \quad (9)$$

and “ \oplus ” represents the Minkowski sum. The selection of the data set \mathcal{D}_i aims to include points in a ball of radius R_i around any query state, to make predictions better for points in the frontiers between partitions.

We call the prediction rule $f^*(\cdot)$ smoothed projected kinky inference (SPKI).

Note that the cardinality of the set \mathcal{D}_i is in general much lower than the cardinality of the original data set \mathcal{D} , and hence obtaining a prediction is less cumbersome. The radius R_i must be chosen appropriately, depending on the density of the data set, increasing it to ensure a minimum number of data points in \mathcal{D}_i . The Hölder constant is kept the same for every partition data set.

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.

The partition can be done using machine learning techniques such as the K-means algorithm for clustering or performing principal component analysis, useful given the recursive characteristic of the NARX model. Some references regarding this topic may be found in [5], [15]. In the case study presented in Section V, the set of the partitions are hypercubes, that is, they are done independently through some dimensions of $\mathcal{W} \subset \mathbb{R}^{n_w}$.

In what is to follow, we will describe this particular approach in greater detail and analyse the computational complexity. To this end, we modify the prediction to (with high probability) only depend on a finite number of data points that does not grow with number of data points while avoiding loss of consistency.

For simplicity, assume the sample inputs were drawn i.i.d. and are contained in a hypercube $H = \{w : w_i \in [0, 1], \forall i = 1, \dots, d\}$ (once the samples are drawn, we can always ensure this by appropriate rescaling of the workspace). We partition H into m^d sub-hypercubes H_i indexed by the tuple $i := (i_1, \dots, i_d)$. where $i_j \in \{1, \dots, m\}$ ($j = 1, \dots, d$). We define

$$H_{i_j} = \left\{ w : i_j = \min\{m, \max\{1, \lfloor w_j m \rfloor + 1\}\}, \forall j = 1, \dots, d \right\}.$$

3) *Computational complexity*: For simplicity, for now, assume predictions are based exclusively on one of the hypercubes H_i , i.e. $H_i = \mathcal{W}_i$. We will consider the following question: Given a fixed partition of H into m^d hypercubes and assuming we have drawn a sample of $N = |\mathcal{D}|$ data points i.i.d. uniformly from H , what is the computational complexity of making a prediction? That is, how many basic computational steps $T_{N,m}$ are required to compute i and subsequently $f^*(q; \theta, \mathcal{D})$ for any given query $q \in \mathcal{W}$?

To answer this question, we note that, for a given query point $q \in H_i$ computing the index $i = (i_1, \dots, i_d)$ can be done in $\mathcal{O}(d)$ basic computational steps (including potential rescaling if necessary). Once i is determined, we can compute the prediction $f^*(q; \theta, \mathcal{D}_i)$ in $\mathcal{O}(N_i)$ computational steps, where N_i is the number of samples contained in $\mathcal{D}_i = \mathcal{W}_i$. So, predicting the value of the query can be done in at most

$$T_{N,m} := aN_i + bd + c \in \mathcal{O}(N_i + d)$$

computational steps for some (algorithm-dependent) parameters a, b, c .

Now, N_i is a random variable whose distribution depends on the number of cubes m^d and the number of samples $N := N_{\mathcal{D}}$. Assume the data contains N inputs (x_1, \dots, x_N) . Let $E_{i,k} = \{(x_1, \dots, x_N) : |\{x_1, \dots, x_N\} \cap H_i| = k\}$ be the event that hypercube H_i contains exactly $N_i = k$ sample points and let $p := \Pr[H_i] = m^{-d}$. We consider this situation as a sequence of N Bernoulli trials with p being the success probability. Given our assumption of i.i.d. uniformity, $\Pr[E_{i,k}] = \binom{N}{k} p^k (1-p)^{N-k}$. Therefore, the probability that N_i is within some desired interval $[\underline{k}, \bar{k}]$ is

$$\Pr[N_i \in \{\underline{k}, \dots, \bar{k}\}] = \sum_{k=\underline{k}}^{\bar{k}} \binom{N}{k} p^k (1-p)^{N-k}.$$

In summary, to answer our first question, given a partition into m^d hypercubes, we have

$$\Pr[T_{N,m} \in \{a\underline{k} + bd + c, \dots, a\bar{k} + bd + c\}] \quad (10)$$

$$= \sum_{k=\underline{k}}^{\bar{k}} \binom{N}{k} m^{-dk} (1 - m^{-d})^{N-k}. \quad (11)$$

With N_i following a binomial distribution, the expected value of the computational effort is

$$\mathbb{E}[T_{N,m}] = apN + bd + c \quad (12)$$

$$= am^{-d}N + bd + c \in \mathcal{O}(m^{-d}N + d). \quad (13)$$

These results can be utilised to answer the question of how to choose parameters m to control the average and maximal computational effort per query point. This is of particular importance in MPC where predictions are made repeatedly during an optimisation process that needs to terminate within a given time span that is smaller than the sampling period length. Of course, in practise, we may furthermore cap the per query prediction time by organising the data in \mathcal{D}_i in a list and only utilising the data of that list to make a prediction that can be processed within the prescribed computational budget.

In our analysis, we have assumed that we only take H_i into account when computing \mathcal{D}_i . To facilitate smoothness of the prediction surface however, we also consider a selection of adjacent hypercubes. In the full setting we include samples from all neighbouring hypercubes in the set $\{H_{i+j} | j \in \{-1, 0, 1\}^d\}$. Of course, this means that the computational complexity increases by a factor of $\mathcal{O}(3^d)$. If one desires to avoid this exponential dependence on d , for high-dimensional problems in particular, it may instead be desirable to achieve a compromise by only include those neighbouring hypercubes that share a face with H_i . That is, we only take into account samples from the union of the hypercubes in the set $\{H_{i+j} | j \in \{-1, 0, 1\}^d \wedge \|j\|_1 \leq 1\}$. Since the latter set only contains $\mathcal{O}(d)$ hypercubes, with this choice, the overall complexity increases merely by a factor that grows linearly with the input space dimensionality d .

As a final remark, it should be emphasised that several extensions of our analysis might have to be made to adjust it to better match conditions that might arise in reality. Firstly, the collected data might not have been collected i.i.d. from an uniform distribution. Indeed, the distribution might be unknown (or distributional assumption questionable). In that case, an approach would be to fit a distribution to the collected data via standard density estimation techniques. In that case, our hypercubes will have varying success probabilities p that have to be taken into account when computing the expected computational effort and the probabilistic bounds. Secondly, the inputs may follow a different distribution or none at all. In that case, depending on the case, one should marginalise over the input distribution or might resort to worst-case analysis.

IV. DATA-BASED PREDICTIVE CONTROL

This section presents a model predictive controller designed to ensure robust stability of the real system in closed-loop considering soft constraints in the outputs and hard constraints in the inputs. In the proposed controller a prediction horizon larger than the control horizon has been considered in order to enhance the closed-loop performance and to increase the domain of attraction. This is particularly interesting when the system to be controlled is nonlinear.

For a given state of the plant and a sequence of future control actions (\hat{u}_0, \dots) at sampling time k , the state prediction model to be used in the MPC is derived from the prediction function (7) as follows:

$$\begin{aligned} \hat{x}(j+1|k) &= \hat{F}(\hat{x}(j|k), \hat{u}(j); \theta, \mathcal{D}) \\ \hat{y}(j|k) &= M\hat{x}(j|k) \end{aligned}$$

where the predicted state

$$\begin{aligned} \hat{x}(j|k) &= (\hat{y}(j|k), \dots, \hat{y}(1|k), y(k), \dots, y(k+j-n_a), \\ &\quad \hat{u}(j-1), \dots, \hat{u}(0), \dots, u(j-n_b)), \end{aligned}$$

includes real measurements y or u if $n_a \geq j$ or $n_b > j$, respectively, and only estimated values \hat{y} or \hat{u} otherwise.

The model is:

$$\begin{aligned} \hat{F}(\hat{x}(j|k), \hat{u}(j); \theta, \mathcal{D}) &= (\mathbf{f}^*(\hat{x}(j|k), \hat{u}(j); \theta, \mathcal{D}), \\ &\hat{y}(j|k), \dots, y(k), \dots, \\ &y(k+j-n_a+1), \\ &\hat{u}(j), \dots, \hat{u}(j-n_b+1)) \end{aligned}$$

and the output matrix is $M = [I_{n_y}, 0, \dots, 0] \in \mathbb{R}^{n_x}$, being I_{n_y} the identity matrix.

To derive the predictive controller, a positive definite stage cost function $\ell(y, u)$ is defined as:

$$\ell(y, u) = \ell_t(y - y_r, u - u_r) + \ell_b(y)$$

where the term $\ell_t(y - y_r, u - u_r)$ penalizes the tracking error of inputs and outputs w.r.t. the reference set-point given by (y_r, u_r) and $\ell_b(y)$ is a barrier function that implements the soft constraint on the outputs. The barrier cost function $\ell_b(\cdot)$ must satisfy that $\ell_b(y) = 0$ for all $y \in \mathcal{Y}$ and there exists two \mathcal{K} functions α_b and β_b such that

$$\alpha_b(\mathfrak{d}(y, \mathcal{Y})) \geq \ell_b(y) \geq \beta_b(\mathfrak{d}(y, \mathcal{Y})) \quad \forall y$$

Assumption 1: There exist two \mathcal{K} -functions α_y and α_u such that

$$\ell(y, u) \geq \alpha_y(\|y - y_r\|) + \alpha_u(\|u - u_r\|).$$

Besides, $\ell(y, u)$ is assumed to be continuous.

The optimisation problem $P_{N_c, N_p}(x(k), \mathcal{D})$ to be solved is the following:

$$\begin{aligned} \min_{\hat{u}} \quad & J_{N_c, N_p}(x(k), \hat{u}) = \sum_{i=0}^{N_c-1} \ell(\hat{y}(i|k), \hat{u}(i)) \\ & + \sum_{i=N_c}^{N_p-1} \ell(\hat{y}(i|k), \kappa_f(\hat{x}(i|k))) \\ & + \lambda V_f(\hat{x}(N_p|k) - x_r) \end{aligned} \quad (14a)$$

$$\text{s.t.} \quad \hat{x}(0|k) = x(k) \quad (14b)$$

$$\hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k), \hat{u}(j); \theta, \mathcal{D}), \quad j \in \mathbb{I}_0^{N_p-1} \quad (14c)$$

$$\hat{y}(j|k) = M\hat{x}(j|k) \quad (14d)$$

$$\hat{u}(j) \in \mathcal{U} \quad (14e)$$

where $\lambda \geq 1$ is a design parameter of the controller, N_p is the prediction horizon, and $N_c \leq N_p$ is the control horizon. Asymptotic stability is ensured by designing a suitable terminal cost function $V_f(\cdot)$ and a terminal control law $\kappa_f(\cdot)$. Notice that no terminal constraint is considered since the terminal cost function V_f is weighted by a factor $\lambda \geq 1$, following the procedure proposed in [9], and extended for a prediction horizon larger than the control horizon.

The MPC control law is $u(k) = \kappa_{\text{MPC}}(x(k), y_r; \theta, \mathcal{D}) = \hat{u}^*(0)$, which depends on the regressors of the system and the collected historical data set \mathcal{D} .

A. Stability analysis of the controller

In order to demonstrate the stability of the proposed controller, first the asymptotic stability of the *nominal case* will be proven, that is, when the proposed controller is used to regulate a system whose model is equal to the prediction one (5).

Theorem 1: There exists a terminal controller $\kappa_f(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$, a terminal cost function $V_f(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ and a region $\Omega_\beta = \{x : V_f(x) \leq \beta\}$, where β is a positive constant such that for all $x \in \Omega_\beta$:

$$\begin{aligned} \alpha_1(\|x - x_r\|) &\leq V_f(x) \leq \alpha_2(\|x - x_r\|), \\ V_f(x^+) - V_f(x) &\leq -\ell_t(y - y_r, \kappa_f(x) - u_r), \\ Mx &\in \mathcal{Y}, \\ \kappa_f(x) &\in \mathcal{U}. \end{aligned}$$

Consider the region

$$X_{N_c, N_p}(\lambda) = \left\{ x : J_{N_c, N_p}^*(x(k)) \leq (N_p - n)\gamma + \lambda\beta \right\},$$

where γ is a positive constant and $n = \max(n_a, n_b)$.

Then, given $N_p \geq n$, for all $\lambda \geq 1$ and all $x(0) \in X_{N_c, N_p}(\lambda)$, the closed-loop system $x^+ = \hat{F}(x, \kappa_{\text{MPC}}(x; \theta, \mathcal{D}); \theta, \mathcal{D})$ is asymptotically stable.

Proof:

Notice that if $x \in \Omega_\beta$, $y \in \mathcal{Y}$ so $\ell_b = 0$, and thus $\ell(y, u) = \ell_t(y - y_r, u - u_r)$. For the sake of clarity it may be used x_j or $x(j|k)$ indistinctly. The proof uses the following three lemmas:

Lemma 1: If the assumptions of Theorem 1 hold, then $J_{N_c, N_p}^*(x(k)) \leq \lambda V_f(x(k)) \quad \forall x(k) \in \Omega_\beta$.

Proof: for any system controlled by $u(j|k) = \kappa_f(x(j|k))$

$$V_f(x(j|k)) - V_f(x(j+1|k)) \geq \ell(y(j|k), \kappa_f(x(j|k))).$$

Summing for the whole sequence:

$$\begin{aligned} \sum_{j=0}^{N_p-1} V_f(x(j|k)) - V_f(x(j+1|k)) \\ = V_f(x(k)) - V_f(x(N_p|k)) \geq \sum_{j=0}^{N_p-1} \ell(y(j|k), \kappa_f(x(j|k))) \end{aligned}$$

and by optimality, provided that $\lambda \geq 1$, $\forall N_p \geq N_c \geq 1$

$$\begin{aligned} \lambda V_f(x(k)) &\geq \sum_{j=0}^{N_p-1} \ell(y(j|k), \kappa_f(x(j|k))) + \lambda V_f(x(N_p|k)) \\ &\geq J_{N_c, N_p}^*(x(k)). \end{aligned} \quad (15)$$

Lemma 2: If $x^*(j|k) \notin \Omega_\beta$, $\forall j = 0, \dots, N_p - 1$, then

$$J_{N_c, N_p}^*(x(k)) \geq (N_p - n)\gamma + \lambda\beta.$$

Proof: let define $\ell_{\text{ext}}(x, u)$ as the stage cost extended to consider a positive definite function of the whole state:

$$\ell_{\text{ext}}(x(k), u(k)) = \frac{1}{n} \sum_{i=0}^n \ell(y(k-n+i), u(k-n+i)),$$

where $n = \max(n_a, n_b)$, and with a slightly abuse of notation, the terms $y(k - n + i)$ and $u(k - n + i)$ are set to y_r and u_r if $y(k - n + i)$ or $u(k - n + i)$ are not included in $x(k)$, respectively.

Then γ is defined as a positive constant such that $\forall x \notin \Omega_\beta$, $\ell_{\text{ext}}(x, u) \geq \gamma$. Hence,

$$\begin{aligned} J_{N_c, N_p}^*(x(k)) &= \sum_{j=0}^{N_p-1} \ell(y^*(j), u^*(j)) + \lambda V_f(x^*(N_p)) \\ &\geq \sum_{k=0}^{\lfloor \frac{N_p}{n} \rfloor - 1} \sum_{i=0}^n \ell(y^*(k + n - i), u^*(k + n - i)) + \lambda V_f(x^*(N_p)) \\ &\geq \left\lfloor \frac{N_p}{n} \right\rfloor n\gamma + \lambda\beta \\ &\geq (N_p - n)\gamma + \lambda\beta. \end{aligned}$$

Lemma 3: If the assumptions of Theorem 1 hold, and if $x^*(N_p|k) \notin \Omega_\beta$, then $x^*(j|k) \notin \Omega_\beta$, $\forall j = 0, \dots, N_p - 1$.

Proof: it is proven by contradiction: (i) assume that there exists any instant $i < N_c$ in which $x_i^* \in \Omega_\beta$. From Lemma 1 it is inferred that for a problem of horizons $N_c - i$, $N_p - i$

$$\lambda V_f(x_i^*) \geq J_{N_c - i, N_p - i}^*(x_i^*)$$

From Bellman's optimality principle, and provided that $\ell(y, u)$ is positive definite, it is derived that

$$J_{N_c - i, N_p - i}^*(x_i^*) \geq \lambda V_f(x_{N_p}^*)$$

and given that $x_{N_p}^* \notin \Omega_\beta$, then

$$\lambda V_f(x_{N_p}^*) > \lambda\beta$$

Hence $\lambda V_f(x_i^*) > \lambda\beta$, which is a contradiction, so $x_i^* \notin \Omega_\beta$.

(ii) Assume there exists any instant $i \geq N_c$ in which $x_i^* \in \Omega_\beta$, then $x_{j+1}^* = \hat{F}(x_j^*, k_f(x_j^*))$ for $j = i, \dots, N_p$. Since Ω_β is a positive invariant set for this system, we have that $x_j^* \in \Omega_\beta$ for $j = i, \dots, N_p$, leading again to a contradiction. ■

Given these three lemmas, we have that $\forall x \in X_{N_c, N_p}(\lambda)$, if $x_{N_p}^* \notin \Omega_\beta$, then $x_j^* \notin \Omega_\beta \forall j = 0, \dots, N_p - 1$, and therefore

$$J_{N_c, N_p}^*(x(k)) \geq (N_p - n)\gamma + \lambda\beta.$$

So on the contrary, if $J_{N_c, N_p}^*(x(k)) \leq (N_p - n)\gamma + \lambda\beta$, then $x_{N_p}^* \in \Omega_\beta$. Next, we are ready to prove the statement of the theorem. First, recursive feasibility is proven:

Assume that $x(k) \in X_{N_c, N_p}(\lambda)$, then $x_{N_p}^* \in \Omega_\beta$. Define the sequence of future control inputs for $x(k+1)$ as

$$\tilde{u}(j|k+1) = \begin{cases} u^*(j|k) & j = 0, \dots, N_c - 1 \\ \kappa_f(\tilde{x}(j|k+1)) & j = N_c, \dots, N_p - 1 \end{cases}$$

and define $\tilde{x}(j|k+1)$ as the predicted trajectory. Then, from standard arguments of MPC proofs, this sequence is feasible for the optimisation problem since $u^*(k)$ is feasible. Besides, the difference between the costs associated to these trajectories is:

$$\begin{aligned} &\tilde{J}_{N_c, N_p}(x(k+1)) - J_{N_c, N_p}^*(x(k)) \\ &= -\ell(y(k), u^*(k)) + [\ell(\tilde{y}(N_p|k+1), \kappa_f(\tilde{x}(N_p|k+1))) + \\ &\quad + \lambda V_f(\tilde{x}(N_p|k+1)) - \lambda V_f(x^*(N_p|k))]. \end{aligned}$$

The sum among brackets is negative, since $x_{N_p}^* \in \Omega_\beta$, thus

$$\tilde{J}_{N_c, N_p}(x(k+1)) - J_{N_c, N_p}^*(x(k)) \leq -\ell(y(k), u^*(k)),$$

and by optimality

$$J_{N_c, N_p}^*(x(k+1)) \leq \tilde{J}_{N_c, N_p}(x(k+1))$$

$$J_{N_c, N_p}^*(x(k+1)) - J_{N_c, N_p}^*(x(k)) \leq -\ell(y(k), u^*(k)) \quad (16)$$

So since $x(k) \in X_{N_c, N_p}(\lambda)$ implies that $J_{N_c, N_p}^*(x(k)) \leq (N_p - n)\gamma + \lambda\beta$, and then $J_{N_c, N_p}^*(x(k+1)) \leq (N_p - n)\gamma + \lambda\beta$, which implies that $x(k+1) \in X_{N_c, N_p}(\lambda)$. Therefore the closed-loop system is recursively feasible.

Then, asymptotic stability is proven. Given the definition of $x(k)$ in (2),

$$\begin{aligned} \sum_{j=k-n}^k \ell(y(j), u(j)) &\geq \sum_{j=0}^n \alpha_y(\|y(k-j)\|) + \alpha_u(\|u(k-j)\|) \\ &\geq \alpha_x(\|x(k)\|) + \alpha_u(\|u(k)\|) \\ &\geq \alpha_x(\|x(k)\|), \end{aligned} \quad (17)$$

where $n = \max(n_a, n_b)$ and α_y, α_u , and α_x are \mathcal{K} functions. Defining

$$W(x_k) = \sum_{j=0}^n J_{N_c, N_p}^*(x(k-j)),$$

we can infer that

$$\begin{aligned} &W(x(k+1)) - W(x(k)) \\ &= J_{N_c, N_p}^*(x(k+1)) - J_{N_c, N_p}^*(x(k-n)) \\ &\leq -\sum_{j=k-n}^k \ell(y(j), u(j)) \leq -\alpha_x(\|x(k) - x_r\|) \end{aligned}$$

where the first inequality is derived by recursion of (16), and the second inequality comes from (17). Besides, $W(x) \geq$

$\sum_{j=k-n}^k \ell(y(j), u(j)) \geq \alpha_x(\|x - x_r\|)$ and, by Prop. B.25 in [16], there exists a certain \mathcal{K} function $\alpha_2(\cdot)$ such that $W(x) \leq \alpha_2(\|x - x_r\|)$. Hence $W(x(k))$ is a Lyapunov function of the system, which is asymptotically stable. ■

Remark 3: Whereas N_p and λ are design parameters, γ and β are problem-dependent parameters. They require solving three global optimization problems, whose ingredients are sketched below.

According to the definition of β in Theorem 1, let us define β_1 as:

$$\beta_1 = \arg \min_x V_f(x) \quad (18)$$

$$\text{s.t. } \lambda V_f(\hat{F}(x, \kappa_f(x); \theta, \mathcal{D})) - \lambda V_f(x) \geq -\ell(Mx, \kappa_f(x))$$

It must also be guaranteed that for all $V_f(x) \leq \beta$, $\kappa_f(x) \in \mathcal{U}$, so we define

$$\begin{aligned} \beta_2 = \max_{\beta} \quad & \beta \\ \text{s.t.} \quad & \kappa_f(x) \in \mathcal{U} \\ & Mx \in \mathcal{Y}, \forall x \in \{x : V_f(x) \leq \beta\} \end{aligned} \quad (19)$$

Then $\beta = \min(\beta_1, \beta_2)$. γ is obtained as:

$$\begin{aligned} \gamma = \min \quad & \ell_{\text{ext}}(x, \kappa_f(x)) \\ \text{s.t.} \quad & V_f(x) \geq \beta. \end{aligned} \quad (20)$$

Note that these calculations are done offline.

Corollary 1: Under the assumptions of Theorem 1, it can be proven that:

- (i) $X_{N_c, N_p}(\lambda_1) \subseteq X_{N_c, N_p}(\lambda_2) \forall \lambda_1 \leq \lambda_2$
- (ii) $X_{N_c, N_1}(\lambda) \subseteq X_{N_c, N_2}(\lambda) \forall N_1 \leq N_2$

Proof: first it is proven that if $\lambda_1 \leq \lambda_2$ and $x \in X_{N_c, N_p}(\lambda_1)$, then $x \in X_{N_c, N_p}(\lambda_2)$. Define \hat{u}^* , \hat{x}^* and \hat{J}_{N_c, N_p}^* as the optimal solution for $x \in X_{N_c, N_p}(\lambda_1)$ and \tilde{u}^* , \tilde{x}^* and \tilde{J}_{N_c, N_p}^* as the optimal solution for $x \in X_{N_c, N_p}(\lambda_2)$. Taking into account that \hat{u}^* is a feasible solution of the optimisation problem for λ_2 ,

$$\begin{aligned} \tilde{J}_{N_c, N_p}^*(x) &= \sum_{i=0}^{N_p-1} \ell(\tilde{x}_i^*, \tilde{u}_i^*) + \lambda_2 V_f(\tilde{x}_{N_p}^*) \\ &\leq \sum_{i=0}^{N_p-1} \ell(\hat{x}_i^*, \hat{u}_i^*) + \lambda_2 V_f(\hat{x}_{N_p}^*) \\ &= \sum_{i=0}^{N_p-1} \ell(\hat{x}_i^*, \hat{u}_i^*) + \lambda_1 V_f(\hat{x}_{N_p}^*) + \lambda_2 V_f(\hat{x}_{N_p}^*) - \lambda_1 V_f(\hat{x}_{N_p}^*) \\ &= \hat{J}_{N_c, N_p}^*(x) + (\lambda_2 - \lambda_1) V_f(\hat{x}_{N_p}^*) \\ &\leq \hat{J}_{N_c, N_p}^*(x) + (\lambda_2 - \lambda_1) \beta \\ &\leq (N_p - n) \gamma + \lambda_1 \beta + (\lambda_2 - \lambda_1) \beta \\ &= (N_p - n) \gamma + \lambda_2 \beta \end{aligned}$$

and hence $x \in X_{N_c, N_p}(\lambda_2)$. \blacksquare

Now it is proven that if $N_1 \leq N_2$ and $x \in X_{N_c, N_1}(\lambda)$, then $x \in X_{N_c, N_2}(\lambda)$.

Provided that $J_{N_c, N_1}^*(x(k)) \leq (N_1 - n) \gamma + \lambda \beta \forall x \in X_{N_c, N_1}(\lambda)$ and $N_1 \leq N_2$, from (15) it is derived that

$$V_f(x_{N_1}^*) \geq \sum_{j=N_1}^{N_2-1} \ell(y_j^*, \kappa_f(x_k^*)) + V_f(x_{N_2}^*)$$

which implies that $J_{N_c, N_1}^* \geq J_{N_c, N_2}^*$, so

$$J_{N_c, N_2}^* \leq J_{N_c, N_1}^* \leq (N_1 - n) \gamma + \lambda \beta \leq (N_2 - n) \gamma + \lambda \beta$$

and hence $X_{N_c, N_1} \subseteq X_{N_c, N_2} \forall N_1 \leq N_2$. \blacksquare

Remark 4: From standard theory of MPC, it is well known that increasing the control horizon N_c increases the domain of attraction $X_{N_c, N_p}(\lambda)$, with the respective increase in the number of decision variables. Corollary 1 provides other two ways of increasing the domain of attraction of the

MPC: (i) increasing the weighting factor λ and (ii) increasing the prediction horizon N_p . A larger weighting factor provides a larger domain of attraction, but the performance might be worse. On the other hand, a larger prediction horizon does not only increase the domain of attraction, but it also improves the performance, since $J_{N_c, N_2}^* \leq J_{N_c, N_1}^*$. The number of decision variables does not increase with N_p , but it increases the number of constraints in the optimisation problem, which might lead to a mild increase of the computational cost.

Once the nominal asymptotic stability is proven, then it is shown that the real plant controlled with the proposed predictive control law is input-to-state stable w.r.t. the model mismatches under the following assumption.

Assumption 2: It is assumed that the radius R_i in equation (9) is large enough to ensure that the function $\mathfrak{f}^*(\cdot, \cdot)$ is Hölder continuous.

Definition 1: A system $x^+ = f(x) + \nu$ is input-to-state stable (ISS) w.r.t. ν if there exists a \mathcal{KL} function β and a \mathcal{K} function α such that $\|x(k)\| \leq \beta(\|x(0)\|, k) + \sup_{j \in [0, k]} \alpha(\|\nu(j)\|)$.

Now, the stability result is stated in the following theorem:

Theorem 2: Assume that the stage cost function $\ell(\cdot, \cdot)$, the terminal control law $\kappa_f(\cdot)$ and the terminal cost function $V_f(\cdot)$ are uniformly continuous functions in the feasibility region $X_{N_c, N_p}(\lambda)$. Assume that the hypotheses of Theorem 1 and Assumption 2 hold. Then real plant, given by the system (1), controlled by $u(k) = \kappa_{\text{MPC}}(x(k), y_r; \theta, \mathcal{D})$ is input-to-state stable w.r.t. the model mismatch signal:

$$d(k) = f(x(k), u(k)) - \mathfrak{f}^*(x(k), u(k); \theta, \mathcal{D})$$

Proof: First, notice that the model of real plant can be posed as $f(x(k), u(k)) = \mathfrak{f}^*(x(k), u(k)) + d(k)$ and the signal $d(k)$ is bounded by μ . Given that the function $\mathfrak{f}^*(\cdot, \cdot)$ is Hölder continuous, the model function $F(\cdot, \cdot)$ is also Hölder continuous. Since the stage cost function $\ell(\cdot, \cdot)$ and the terminal cost function $V_f(\cdot)$ are continuous functions and $X_{N_c, N_p}(\lambda)$ is a compact set, then they are uniformly continuous. From Proposition 1, case C1, in [17] it is inferred that the closed-loop system is ISS w.r.t. the signal $d(k)$ that describes the model mismatch. \blacksquare

Remark 5: This theorem resorts on Assumption 2, which may be difficult to demonstrate. Notice that the assumption can always be satisfied since for $\mathcal{D}_i = \mathcal{D}$ the model function is Hölder continuous. In case one cannot ensure Hölder continuity, stability is not lost. Instead, according to the definition of input-to-state practical stability (ISpS, see Definition 6 in [17]), it can be proven that if the discontinuity jump is upper bounded by c , then

$$\|\mathfrak{f}^*(w + \Delta w, u) - \mathfrak{f}^*(w, u)\| \leq \sigma(c) + \hat{L} \|\Delta w\|^\alpha$$

where σ is a \mathcal{K} function. Using similar arguments to the proof of Theorem 2 and Definition 6 in [17], it can be proven that the closed-loop system is ISpS, since there exists a \mathcal{KL} function β and \mathcal{K} functions α and ω such that:

$$\|x(k)\| \leq \beta(\|x(0)\|, k) + \sup_{j \in [0, k]} \alpha(\|d(j)\|) + \omega(c)$$

Remark 6: The calculation of the control action requires the solution of a mathematical programming problem on line. This is typically a gradient-based algorithm that may be sensitive to abrupt variations of the gradients, which may be derived from discontinuity of the derivative of the models. As it is illustrated in the case study, this may produce convergence issues leading to spikes in the evolution of the control action. To avoid this effect, the tracking stage cost function can be modified adding a term penalizing the variation of the control input, i.e. $\|\hat{u}(i) - \hat{u}(i-1)\|_S^2$. The resulting tracking stage cost can be recasted, w.l.o.g. as a function of the state (that is, the regressor), $\ell_t(x-x_r, u-u_r)$, provided that $n_b \geq 1$. The stability analysis presented can be extended to this case.

V. CASE STUDY

In this Section we consider the control of a continuously-stirred tank reactor presented in [18]. The experiments carried out to generate the data sets and the procedure to tune the proposed controller are described in detail.

The input of the reactor is the reference of the coolant temperature T_r (K) and the output is the concentration of the reactant, C_A (mol/l), in the reaction $A \rightarrow B$. The temperature of the tank and the coolant are given by T and T_c respectively (K). It is assumed that the evolution of the plant is given by the following set of differential equations:

$$\begin{aligned} \frac{dC_A(t)}{dt} &= \frac{q_0}{V} \cdot (C_{Af} - C_A(t)) - k_0 \cdot e^{\left(-\frac{E}{R \cdot T(t)}\right)} \cdot C_A(t) \\ \frac{dT(t)}{dt} &= \frac{q_0}{V} \cdot (T_f - T(t)) + \\ &\quad + \frac{(-\Delta H_r) \cdot k_0}{\rho \cdot C_p} \cdot e^{\left(-\frac{E}{R \cdot T(t)}\right)} \cdot C_A(t) + \\ &\quad + \frac{U \cdot A}{V \cdot \rho \cdot C_p} \cdot (T_c(t) - T(t)) \\ \frac{dT_c(t)}{dt} &= \frac{T_c(t) - T_r(t)}{\tau} \end{aligned}$$

The parameters of the model are given in Table I. 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 concentration sensor adds an error of 2.5% of the measurement. 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 no hard constraints in the states or the outputs are considered.

A. Obtaining the data set

The workspace is bounded by $T_r^{\min} = 335 \text{ K}$ and $T_r^{\max} = 372 \text{ K}$. First, the static characteristic is estimated using a sequence of steps in the input from T_r^{\min} to T_r^{\max} with 0.5 K increments, and each step long enough to reach a steady state, since the CSTR is stable. The result is shown in Figure 3. Besides obtaining the equilibrium points of the system, this test is used to adjust the sampling time, which is set to $\tau/20 = 30 \text{ s}$, where τ stands for the mean settling time of the sequence of steps applied.

TABLE I
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

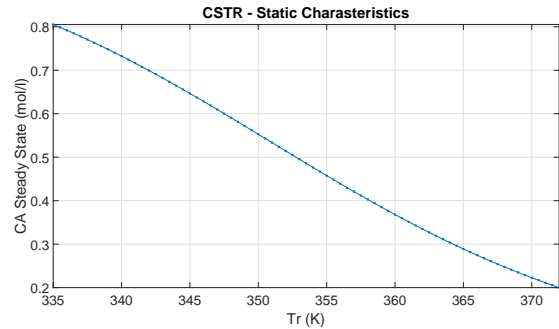


Fig. 3. Static characteristics.

This test is also used to bound the output variable C_A , which is defined by $C_A^{\max} = 0.8053 \text{ mol/l}$ and $C_A^{\min} = 0.2 \text{ mol/l}$ (note that the system has an inverse static characteristic). This restriction is treated by means of a soft constraint using a barrier cost, as it will be explained later. In addition, an equilibrium point is chosen to be the reference operating point, i.e. $C_A^{\text{ref}} = 0.439 \text{ mol/l}$, $T_r^{\text{ref}} = 356 \text{ K}$.

After defining the static characteristic, a set of experiments are carried out to obtain the data for the predictor. The experiments are designed using the methodologies presented in [19] to identify the dynamics of a system within a workspace: a relay test is carried out to estimate the crossover frequency of the system around the operating point [20], with limits T_r^{\max} and T_r^{\min} . The crossover frequency estimated is 3.7 mHz. Figure 4 shows the simulation. Then, a sequence of chirp signals covering the workspace are applied to generate the raw data set containing the trajectories of concentrations and temperatures, \mathcal{D}_{raw} . The parameters of the chirp signals (length of the signal τ_f , initial and final frequency f_0, f_f , center T_{r_0} and amplitude A) are shown in Table II. The simulation is shown in Figure 5,

The points of the static characteristics are added to the data set obtained with the chirp signals (the resulting data points are represented in Figure 6).

In addition, several tests with random input signals are carried out in order to obtain data sets for cross-validation (see Figure 7). The same input signal is applied several times in order to estimate a bound of the measurement error of C_A . This error is estimated as $\bar{\epsilon} = 0.02 \text{ mol/l}$, which is the

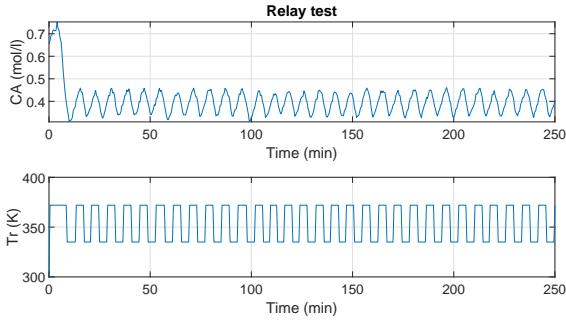


Fig. 4. Relay test response.

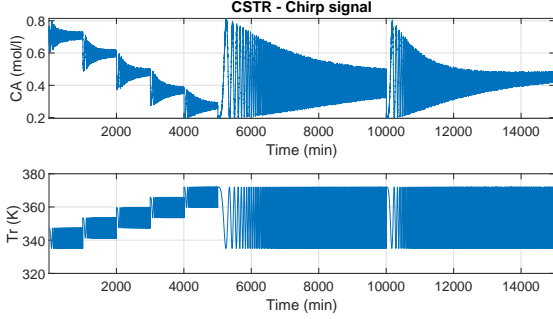


Fig. 5. Chirp signals used to obtain the data set.

expected value given the 2.5% measurement error used to generate the simulation.

In order to implement the prediction and optimisation problem, it is desirable that the input-output variables are suitably scaled, so every component is expressed in the range $[0,1]$. The data set \mathcal{D}_{raw} is 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):

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

$$y = \frac{C_A - C_A^{\min}}{C_A^{\max} - C_A^{\min}}$$

To obtain the SPKI model, the parameters n_a and n_b of the NARX regressor have to be chosen. To this end, for various values of these parameters, the SPKI predictor is designed and its corresponding validation error is computed, to choose the best set of parameters.

For each set of parameters, a data set $\mathcal{D} = \{z_k, w_k\}$, with $k = 1 \dots N_{\mathcal{D}}$, is obtained from the experimental data as explained in Section III.

Thus, for various combinations of the memory horizons, the Hölder parameters of the system are obtained using (4). Using the obtained constant and exponent, the SPKI predictor can be defined as in (8).

The SKI filtering of equation (7) is done predicting with $\sigma_0 = 2/(n_w + 2)$, $\sigma_i = 1/(n_w + 2)$, and increment δ set to 0.015. For this reason the computation time used for

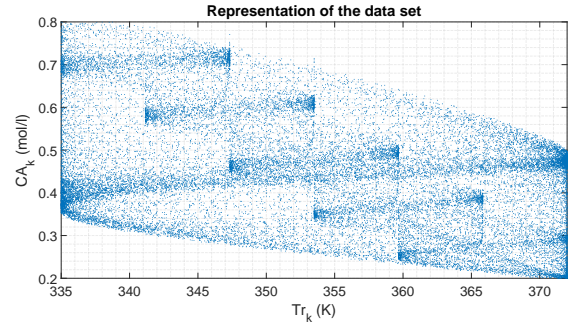


Fig. 6. 2-D representation of \mathcal{D} . Pairs $\{C_{A_k}, T_{r_k}\}$, for $k = 1, \dots, 30000$.

TABLE II
PARAMETERS OF THE CHIRP SIGNALS

Signal	τ_f (min)	f_0 (μHz)	f_f (mHz)	T_0 (K)	A (K)
1	1000	16.6	3.7	341.17	6.1667
2	1000	16.6	3.7	347.33	6.1667
3	1000	16.6	3.7	353.5	6.1667
4	1000	16.6	3.7	359.67	6.1667
5	1000	16.6	3.7	365.83	6.1667
6	5000	1.66	1.7	353.5	18.5
7	5000	1.66	3.7	353.5	18.5

prediction increase linearly with the number of regression terms (n_a, n_b) .

Finally, the data from the other tests is used to calculate the prediction error, as the difference between the real output and the concentration estimated by the predictor, for the same input signal. As it is represented in Figure 8, for $n_a = 2$ and $n_b = 1$, the maximum prediction error is minimized. With these values the Hölder constant results in $L_{\mathcal{D}} = 1.5015$, and the exponent $\alpha = 0.86$. Figure 9 shows the histogram of the validation error for this configuration, whose absolute maximum is $\mu = 0.061$ mol/l.

In order to reduce the computational effort, the regressor space \mathcal{W} is divided into a set of overlapping regions following the procedure presented in Section III-C. Given the recursive characteristic of the NARX model, data points defined by a sequence of temporal signals (e.g. $\{y_k, y_{k-1}\}$) are very concentrated, due to the small variance of the signals between sequential time steps. For this reason, the regions \mathcal{W}_i are not defined in the whole \mathcal{W} space, but instead, only the current values of the output and the input (i.e. y_k, u_k) are taken into account. The vector space (u_k, y_k) is divided into a grid of *squares* of 0.01 side length (this corresponds to dividing each component into 100 partitions, provided that they are both scaled between 0 and 1). Instead of using a ball for the overlapping, to each region, its corresponding data is obtained using also data of neighboring regions (see Figure 10).

The prediction for 100 random query points in an Intel® Core™ i7-6700HQ CPU @ 2.60GHz 12GB RAM without the partitions division takes 41.826 s, while using partition-based approach takes only 0.0717 s, for $n_a = 2$ and $n_b = 1$, with prediction difference between them less than

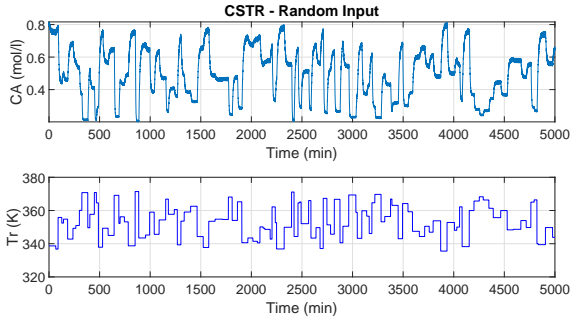


Fig. 7. Test used for cross-validation.

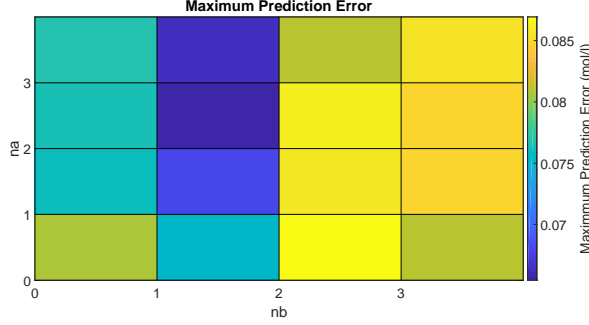


Fig. 8. Dependence of the the prediction error w.r.t. the parameters n_a and n_b .

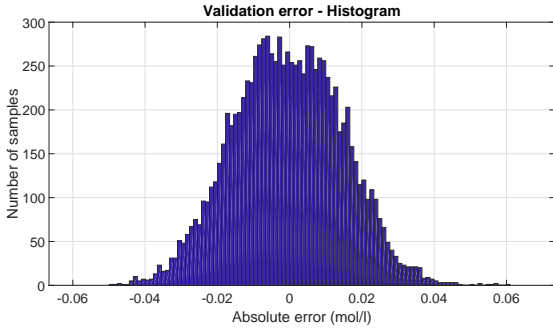


Fig. 9. Histogram of the validation error for $n_a = 2$ and $n_b = 1$.

0.001 mol/l.

B. Control of the reactor

The optimisation problem (14a) is solved in MatLab using the optimisation function `fmincon`. The equations that govern this MPC are those expressed in (14a)-(14e). The reference is set in the equilibrium point of the system given by $\{C_A^{\text{ref}}, T_r^{\text{ref}}\}$.

The stage cost is defined as the sum of a quadratic cost ℓ_t and the barrier cost ℓ_b :

$$\ell_t(x_k, u_k) = \|y_k - y_r\|_Q^2 + \|u_k - u_r\|_R^2 + \|u_k - u_{k-1}\|_S^2$$

$$\ell_b(y_k) = \Upsilon(1 - e^{-\sigma(y_k, \mathcal{V})/\epsilon})$$

The parameters are set to:

$Q = 10$, $R = 5$, $S = 10$ (Remark 6), $\Upsilon = 9999$ and $\epsilon = 3 \times 10^{-3}$. As terminal ingredients, $\kappa_f(x) = K(x - x_r) + u_r$

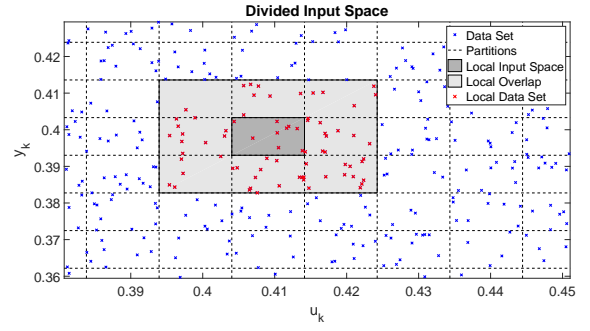


Fig. 10. Partition of the input space \mathcal{W} .

and $V_f(x) = \|x - x_r\|_P^2$ are considered, with $\lambda = 1$. K and P are calculated from the solution of the LQR for a linear model around the reference. This linearized model, calculated numerically from the input-output data, is the following:

$$x_{k+1} = \begin{bmatrix} 0.448 & 0.299 & -0.002 & -0.011 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} -0.0983 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_k$$

This results in:

$$P = \begin{bmatrix} 183.48 & 30.828 & -0.2224 & -1.1638 \\ 30.828 & 14.994 & -0.1079 & -0.5647 \\ -0.2224 & -0.1079 & 0.0008 & 0.0041 \\ -1.1638 & -0.5647 & 0.0041 & 11.021 \end{bmatrix}$$

$$K = [0.4856 \quad 0.2278 \quad -0.0016 \quad -0.0086]$$

Regarding Theorem 1, the parameters γ and β have been computed. Setting the control horizon to $N_c = 4$ and the prediction horizon to $N_p = 14$ results in $\gamma = 1.2046$ and $\beta = 0.0294$. If $x(0) = [0.6 \ 300 \ 300]^T$, $J_{N_c, N_p}^*(x(0)) = 13.721$, so $x(0) \in X_{N_c, N_p}(\lambda)$. Regarding Theorem 2, as far as we analysed the outcomes of the predictor, Assumption 1 holds for every query point observed, making $\ell(\cdot, \cdot)$, $\kappa_f(\cdot)$ and $V_f(\cdot)$ uniformly continuous.

Once the controller is designed, a hundred simulations are carried out, with the same initial state but a hundred different realizations of the 2.5% random noise. The following figures represent the output and the control input. Within each subplot, the constraints appear in black, the reference in red, and the 100 trajectories are represented by the gray band, whose mean value is plotted in blue.

A first experiment is carried out in which the model used by the MPC is the real plant (defined by its set of ODEs). The simulations provide a measure of the best closed-loop behavior achievable assuming perfect knowledge of the model. This result is shown in Figure 11.

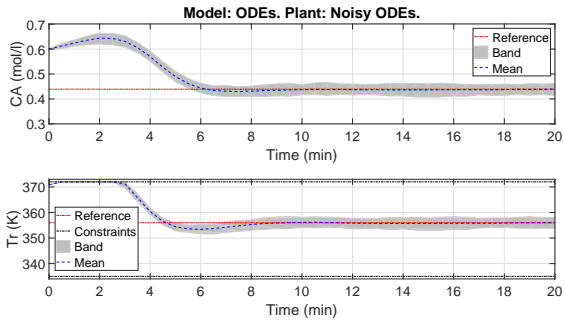


Fig. 11. Ideal ODEs-MPC.

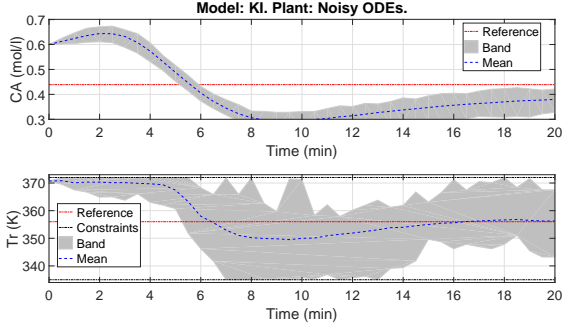


Fig. 12. Primary KI-MPC.

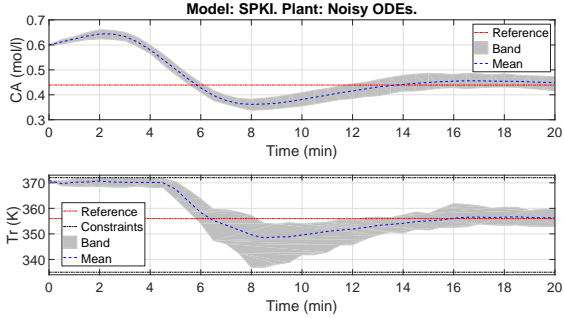


Fig. 13. Data-based SPKI predictive control applied to a CSTR.

Secondly, a MPC controller based on the KI prediction of Section III-A is applied, using the data set \mathcal{D} . Notice that neither the data set was divided into partitions, nor the filtered KI was implemented. The result of this simulation is shown in Figure 12. These results are not satisfactory, since the solver is failing to converge due to the fact that the KI prediction may be non-differentiable or due to the large number of iterations.

Finally, the SPKI MPC presented in this paper is applied, to address the issues mentioned above. The result is shown in Figure 13. The behavior is smoother, with a smaller band than the previous experiment, which indicates greater robustness and a performance closer to the ideal one (ODEs, Figure 11) rather than the simpler KI set.

Apart from that, in order to compare the performance of the presented results, a hundred different simulations of the MPC (with random initial state) are carried out,

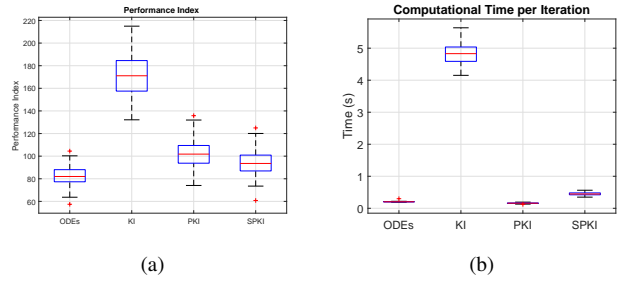


Fig. 14. Boxplot comparison of the different prediction models for 100 simulations of the MPC. (a) represents the performance costs and (b) the computational times.

which results are compared in Figure 14. The boxplots represent maximum, minimum, mean and standard deviation of the performance costs and computational times of the experiments. The performance index of each simulation is defined as:

$$\Phi = \sum_{i=1}^{t_{\text{sim}}} \ell_t(x(i), u(i)) + \ell_b(y(i))$$

VI. CONCLUSIONS

This paper presented a model predictive controller for nonlinear systems based entirely on data. The inference method behind the predictor was based on modified version of the parameter optimised kinky inference, with lower computational effort and smoother predictions. Under the assumption that the model of the system was Hölder continuous, it we have proven that the closed-loop system is input-to-state stable with respect to the prediction errors, considering soft constraints in the output, hard constraints in the input and using different control and prediction horizons and a weighted terminal cost.

The results are demonstrated in a case study of a continuously-stirred tank reactor, for which the whole procedure of obtaining the data and designing the controller is exposed. As it can be observed, with the techniques presented in this paper the performance is improved while the computing times are significantly reduced with respect to simpler configurations of data-based predictive control.

As future work, we suggest to work on extensions to the robust case with hard constraints in the outputs/states. So far, our analysis has been restricted to the offline learning case, where the predictor was inferred from a fixed data set provided before runtime of the controller. In future work, we will investigate extensions to the online learning case where data arrives incrementally during runtime and the control design would have to address exploration-exploitation tradeoffs.

ACKNOWLEDGMENTS

The authors would like to thank the support by the MINECO-Spain and FEDER Funds under project DPI2016-76493-C3-1-R, as well as the VI Plan PPIT-US of the University of Seville.

REFERENCES

- [1] Camacho, E.F., Bordons, C.: ‘Model Predictive Control’. 2nd ed. (Springer-Verlag, 2004)
- [2] Ljung, L.: ‘System identification’. (Springer, 1998)
- [3] Aswani, A., Gonzalez, H., Sastry, S.S., Tomlin, C.: ‘Provably safe and robust learning-based model predictive control’, *Automatica*, 2013, **49**, (5), pp. 1216–1226
- [4] Canale, M., Fagiano, L., Signorile, M.C.: ‘Nonlinear model predictive control from data: a set membership approach’, *International Journal of Robust and Nonlinear Control*, 2014, **24**, (1), pp. 123–139
- [5] Calliess, J.P. ‘Conservative decision-making and inference in uncertain dynamical systems’. University of Oxford, 2014
- [6] Calliess, J.P.: ‘Lipschitz optimisation for lipschitz interpolation’, *American Control Conference (ACC-IEEE)*, 2017, pp. 3141–3146
- [7] Limon, D., Calliess, J., Maciejowski, J.: ‘Learning-based nonlinear model predictive control’, *IFAC-PapersOnLine*, 2017, **50**, (1), pp. 7769–7776
- [8] Manzano, J., Limon, D., Muñoz de la Peña, D., Calliess, J.: ‘Robust data-based model predictive control for nonlinear constrained systems’, *IFAC-PapersOnLine*, 2018, **51**, (20), pp. 505–510
- [9] Limon, D., Alamo, T., Salas, F., Camacho, E.F.: ‘On the stability of MPC without terminal constraint’, *IEEE Transactions on Automatic Control*, 2006, **42**, pp. 832–836
- [10] Leontaritis, I., Billings, S.A.: ‘Input-output parametric models for nonlinear systems part i: deterministic non-linear systems’, *International journal of control*, 1985, **41**, (2), pp. 303–328
- [11] Calliess, J.P.: ‘Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control’, *arXiv preprint arXiv:170100178*, 2016,
- [12] Zabinsky, Z.B., Smith, R.L., Kristinsdottir, B.P.: ‘Optimal estimation of univariate black-box Lipschitz functions with upper and lower bounds.’, *Computers and Operations Research*, 2003,
- [13] Canale, M., Fagiano, L., Signorile, M.C.: ‘Nonlinear model predictive control from data: a set membership approach’, *International Journal of Robust and Nonlinear Control*, 2014, **24**, (1), pp. 123–139
- [14] Milanese, M., Novara, C.: ‘Set membership identification of nonlinear systems’, *Automatica*, 2004,
- [15] Wold, S., Esbensen, K., Geladi, P.: ‘Principal component analysis’, *Chemometrics and intelligent laboratory systems*, 1987, **2**, (1-3), pp. 37–52
- [16] Rawlings, J.B., Mayne, D.Q.: ‘Model Predictive Control: Theory and Design’. 1st ed. (Nob-Hill Publishing, 2009)
- [17] Limon, D., Alamo, T., Raimondo, D.M., de la Peña, D.M., Bravo, J.M., Ferramosca, A., et al. ‘Input-to-state stability: an unifying framework for robust model predictive control’. In: Magni, L., Raimondo, D.M., Allgöwer, F., editors. International Workshop on Assessment and Future Direction of Nonlinear Model Predictive Control. (Springer, 2009. pp. 1–26
- [18] Seborg, D.E., Edgar, T.F., Mellichamp, D.A.: ‘Process Dynamics and Control’. (Wiley, 1989)
- [19] Rivera, D.E., Jun, K.S.: ‘An integrated identification and control design methodology for multivariable process system applications’, *IEEE Control Systems*, 2000, **20**, (3), pp. 25–37
- [20] Åström, K.J., Hägglund, T.: ‘Advanced PID control’. (ISA, 2006)