Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 7769–7776

Learning-based Nonlinear Model Predictive Control ⋆

D. Limon * **J. Calliess** ** **J.M. Maciejowski** **

* *Universidad de Sevilla. dlm@us.es*
** *University of Cambridge. {jpc73,jmm}@cam.ac.uk*

Abstract
This paper presents stabilizing Model Predictive Controllers (MPC) in which prediction models are inferred from experimental data of the inputs and outputs of the plant. Using a nonparametric machine learning technique called LACKI, the estimated (possibly nonlinear) model function together with an estimation of Hölder constant is provided. Based on these, a number of predictive controllers with stability guaranteed by design are proposed. Firstly, the case when the prediction model is estimated off-line is considered and robust stability and recursive feasibility is ensured by using tightened constraints in the optimisation problem. This controller has been extended to the more interesting and complex case: the online learning of the model, where the new data collected from feedback is added to enhance the prediction model. An on-line learning MPC based on a double sequence of predictions is proposed.

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

1. INTRODUCTION

Model-based control design, and particularly Model Predictive Control (MPC), rely on the availability of an accurate description of the plant. When a model of the plant dynamics is unavailable a priori, machine learning methods can be employed to devise such models automatically from observational data.

The objective of this paper is to design a predictive controller based on such learning methods. Here, the learning method should be flexible enough to learn rich classes of dynamical systems, while at the same time, offer bounds on its predictive performance.

The latter is important in the predictive control setting if one wishes to give guarantees on the performance and feasibility of the learning-based controller.

Motivated by these desiderata, previous work on learning-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. In their data-based Nonlinear MPC (NMPC) method, 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 optimization problem happens to be recursively feasible, in spite of the prediction errors.

An alternative approach to learning-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 limited 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 in several ways. Firstly, we propose to utilise nonlinear prediction models based on a recent improvement on NSM learning approach called LACKI Calliess (2016) which estimates Lipschitz constants from the data while still offering worst-case prediction error bounds. Secondly, we design our learning-based NMPC approach in a manner that ensures robust stability and recursive feasibility w.r.t. the estimation error. Thirdly, we also consider the online learning setting where the learning model is updated repeatedly during the runtime of the controller. Using a similar idea to Aswani et al. (2013), two predicted trajectories are entertained in the optimisation problem: one is based on the offline learned model and guarantees the robust constraint satisfaction while the second is calculated by the on-line learning method and it is used for the calculation of the cost func-

⋆ The authors would like to ackowledge to the Spanish MINECO Grant PRX15-00300 and projects DPI2013-48243-C2-2-R and DPI2016-76493-C3-1-R as well as to the Engineering and Physical Research Council, grant no. EP/J012300/1 for funding this work.

10.1016/j.ifacol.2017.08.1050

tion. Since in our case the prediction models are nonlinear and the resulting optimization problem is non-convex, the proof of Aswani et al. (2013) cannot be used for our case. Instead, we have derived input-to-state stability under mild assumptions on the estimator.

The rest of the paper is structured as follows: Following this introduction, a brief rehearsal of some of the learning methods this work could build upon is provided, followed by an introduction to the data-based control problem and a derivation of our predictive controller.

In the first part of the remainder of the paper, the controller is based on a fully trained learning model based on data that has been collected offline. That is, the sample is collected once, before control is attempted, on the basis of which the model is trained, without further updates during the operation of the controller. The LACKI method provides the estimated prediction model together with an estimation of the Lipschitz constant. Based on these ingredients and on the bound of the worst-case estimation error, a NMPC is proposed such that robust stability and recursive feasibility is ensured by design.

In the second part of the paper, we lift this assumption and the case of online learning. Here the sample is repeatedly updated during the runtime of the controller in order to learn a continuously more accurate model of the plant and improve control performance. The previous data-based MPC controller has been extended to the online learning data-based NMPC, which allows us to use a continuously updated prediction model which provides better closed loop performance while robust stability and recursive feasibility are still guaranteed. The proofs have been omitted due to the lack of space. A preliminary version of the proofs can be found in the technical report Limon et al. (2017).

Notation and terminology

Let a, b be two column vectors and (a, b) denote the vector $[a^T, b^T]^T$. Given two sets A, B, the Minkowski sum $A \oplus B$ is defined as the set $\{a + b : a \in A, b \in B\}$ and the Pontryagin difference $A \ominus B$ is defined as the set $\{c : c + b \in A, \forall b \in B\}$. The set of the integers contained in the interval $[a, b]$ with $b \geq a$ is denoted as $\mathbb{I}_{a,b}$. A function $\theta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a $\mathcal{K}$-function if it is strictly increasing and $\theta(0) = 0$. If a $\mathcal{K}$-function is such that $\lim_{s\to\infty} \theta(s) = \infty$ then it is called a $\mathcal{K}_\infty$-function. Remember, a pseudo-metric $\mathfrak{d}_\mathcal{X}(\cdot,\cdot) : \mathcal{X}^2 \to \mathbb{R}$ on a space $\mathcal{X}$ 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 $\mathfrak{d}_\mathcal{X}(x, x') = 0$ even if $x \neq x'$. A mapping $f : \mathcal{X} \to \mathcal{Y}$ between pseudo-metric spaces $(\mathcal{X}, \mathfrak{d}_\mathcal{X})$ and $(\mathcal{Y}, \mathfrak{d}_\mathcal{Y})$ is Hölder continuous with Hölder constant L and exponent p if $\mathfrak{d}_\mathcal{Y}(f(x), f(x')) \leq L\, \mathfrak{d}_\mathcal{X}(x, x')^p, \forall x, x' \in \mathcal{X}$. The smallest Hölder constant of f will be denoted by L^*.

2. LEARNING WITH WORST-CASE PREDICTION BOUNDS

In this work, we will design a controller whose prediction model is inferred (i.e. learned) from data employing machine learning (i.e. regression) techniques. As the controller is intended to come with robustness guarantees, it is important that the predictions of the learned model come with deterministic worst-case prediction error bounds. This limits our choice of learning methods to those for which such bounds are currently known.

A class of learning rules, for which this is the case, and which have been proposed in the context of control, are sometimes referred to as *Kinky Inference* [Calliess (2014)] which encompass Lipschitz Interpolation [Sukharev (1978); Beliakov (2006)] and Nonlinear Set Interpolation [Milanese and Novara (2004)]. These methods will be briefly rehearsed next.

Setting. Let $\mathcal{X}$, $\mathcal{Y}$ be two spaces endowed with (pseudo-) metrics $\mathfrak{d}_\mathcal{X} : \mathcal{X}^2 \to \mathbb{R}_{\geq 0}$, $\mathfrak{d}_\mathcal{Y} : \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$, respectively. Spaces $\mathcal{X}, \mathcal{Y}$ 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. That is, for the input space $\mathcal{X}$, we assume $\mathfrak{d}_\mathcal{X}(x + x', x'' + x') = \mathfrak{d}(x, x''), \forall x, x', x'' \in \mathcal{X}$.

For simplicity, throughout the remainder of this work, we will assume the output space is the canonical Hilbert space $\mathcal{Y} = \mathbb{R}^m$ ($m \in \mathbb{N}$) endowed with the $\mathfrak{d}_\mathcal{Y}(y, y') = \|y - y'\|_\infty, \forall y, y' \in \mathcal{Y}$.

Let $f : \mathcal{X} \to \mathcal{Y}$ be a *target* or *ground-truth* function we desire to learn. For our purposes, learning means regression. That is, we utilise the data to construct a computable function that allows us predict values of the target function at any given input.

Assume that, at time step n, we have access to a *sample* or *data set* $\mathcal{D}_n := \{(s_i, \tilde{f}_i) \,|\, i = 1, \ldots, N_n\}$ containing $N_n \in \mathbb{N}$ (possibly erroneous) sample vectors $\tilde{f}_i \in \mathcal{Y}$ of *target function* f at sample input $s_i \in \mathcal{X}$. The sampled function values are allowed to have interval-bounded *observational error* or *noise* given by a bound $\mathfrak{e} : \mathcal{X} \to \mathbb{R}^m_{\geq 0}$. That is, all we know is that $f(s_i) \in \mathbf{O}_i := [\underline{f}_1(s_i), \overline{f}_1(s_i)] \times \ldots \times [\underline{f}_d(s_i), \overline{f}_d(s_i)]$ where $\underline{f}_j(s_i) := \tilde{f}_{i,j} - \mathfrak{e}_j(s_i)$, $\overline{f}(s_i) := \tilde{f}_{i,j} + \mathfrak{e}_j(s_i)$ and $\tilde{f}_{i,j}$ denotes the jth component of vector $\tilde{f}_i$. The interpretation of these errors depends on the given application and this "noise" may be deterministic or stochastic. For instance, in the context of system identification, the sample might be based on noisy measurements of velocities and it may be due to sensor noise or may capture numerical approximation error, as well as input uncertainty Calliess (2014).

Learning. It is our aim to learn target function f in the sense that, combining prior knowledge about f with the observed data $\mathcal{D}_n$, we infer *predictions* $\hat{\mathfrak{f}}_\mathfrak{n}(x)$ of $f(x)$ at unobserved *query inputs* $x \notin \mathcal{X}_{\mathcal{D}_n}$. Here, $\mathcal{X}_{\mathcal{D}_n} = \{s_i | i = 1 \ldots, N_n\}$ refers to the (not necessarily regular) *grid* of sample inputs. In our context, the evaluation of $\hat{\mathfrak{f}}_n$ is what we refer to as *(inductive) inference* or *prediction*. The entire function $\hat{\mathfrak{f}}_\mathfrak{n}$ that is learned to facilitate predictions is referred to as the *predictor*.

A typical *desideratum* of a good predictor is that it is efficiently *computable* and *converges to the target* (up to the observational error given by $\mathfrak{e}$) in the limit of increasingly dense data sets.

In our context, we will understand a *machine learning algorithm* to implement a computable function that maps a data set $\mathcal{D}_n$ to a $\hat{\mathfrak{f}}_n$ (and possibly an uncertainty estimate function $\hat{\mathfrak{v}}_n$).

In this work we will expand on the basis of the following class of predictors to perform learning as inference over unobserved function values:

Definition 1. (Kinky Inference (KI) rule). Let $\mathbb{R}_\infty := \mathbb{R} \cup \{-\infty, \infty\}$ and $\mathcal{X}$ be some space endowed with a pseudo-metric $\mathfrak{d}_\mathcal{X}$. Let $\underline{B}, \bar{B} : \mathcal{X} \to \mathcal{Y} \subseteq \mathbb{R}_\infty^m$ denote *lower- and upper bound functions*, that can be specified in advance and assume $\underline{B}(x) \leq \bar{B}(x), \forall x \in I \subset \mathcal{X}$ component-wise. Furthermore, let e denote a parameter that specifies a deterministic belief about the true observational error bound $\mathfrak{e}$. Given sample set $\mathcal{D}_n$, we define the predictive functions $\hat{\mathfrak{f}}_\mathfrak{n} : \mathcal{X} \to \mathcal{Y}$, $\hat{\mathfrak{v}}_\mathfrak{n} : \mathcal{X} \to \mathbb{R}_{\geq 0}^m$ to perform inference over function values. For $j = 1, \ldots, m$, their jth output components are given by:

$$\begin{aligned}\hat{\mathfrak{f}}_{\mathfrak{n},\mathfrak{j}}(x) :=& \frac{1}{2}\min\{\bar{B}_j(x), \mathfrak{u}_{n,j}(x; L(n))\} \\ &+ \frac{1}{2}\max\{\underline{B}_j(x), \mathfrak{l}_{n,j}(x; L(n))\}, \\ \hat{\mathfrak{v}}_{\mathfrak{n},\mathfrak{j}}(x) :=& \frac{1}{2}\min\{\bar{B}_j(x), \mathfrak{u}_{n,j}\big(x; L(n)\big)\} \\ &- \frac{1}{2}\max\{\underline{B}_j, \mathfrak{l}_{n,j}\big(x; L(n)\big)\}.\end{aligned}$$

Here, $\mathfrak{u}_n\big(\cdot; L(n)\big), \mathfrak{l}_n\big(\cdot; L(n)\big) : \mathcal{X} \to \mathbb{R}^m$ are called ceiling and floor functions, respectively. Their jth component functions are given by

$$\mathfrak{u}_{n,j}\big(x; L(n)\big) := \min_{i=1,\ldots,N_n} \tilde{f}_{i,j} + L_j(n)\,\mathfrak{d}^p(x, s_i) + e_j(x)$$

and

$$\mathfrak{l}_{n,j}\big(x; L(n)\big) := \max_{i=1,\ldots,N_n} \tilde{f}_{i,j} - L_j(n)\,\mathfrak{d}^p(x, s_i) - e_j(x),$$

respectively. Here $p \in \mathbb{R}, L(n) \in \mathbb{R}^m, \forall n \in \mathbb{N}_0$ and functions $\underline{B}, \bar{B}, e : \mathcal{X} \to \mathbb{R}_\infty^m$ are parameters that have to be specified in advance. To disable restrictions of boundedness, it is allowed to specify the upper and lower bound functions to constants ∞ or $-\infty$, respectively. Function $\hat{\mathfrak{f}}_\mathfrak{n}$ is the predictor that is to be utilised for predicting/inferring function values at unseen inputs. Function $\hat{\mathfrak{v}}_\mathfrak{n}(x)$ is meant to quantify the uncertainty of prediction $\hat{\mathfrak{f}}_\mathfrak{n}(x)$.

Note, that while generally $L(n)$ is a vector, we will simplify our exposition in the remainder of this paper by assuming that all its components are equal or, equivalently, that it is a real number.

To provide an intuition, consider the following special case where we have access to a noise-free sample $\mathcal{D}_n$ and suppose the target f is a real-valued $L^* - p$ Hölder continuous function. Observing the noise-free sample point (s_i, f_i) constrains the set of function values $f(x)$ to the set $\mathbb{S}_i(x) = \{\phi \in \mathcal{Y} | \, \mathfrak{d}_\mathcal{Y}(\phi, f_i) \leq L^* \mathfrak{d}_\mathcal{X}(s_i, x)\}$. Considering a set of sample points $\mathcal{D}_n$, target value $f(x)$ is constrained to lie in the intersection $\mathbb{S}(x) = \cap_{i=1}^{N_n} \mathbb{S}_i(x)$. It is easy to see that the floor and ceiling functions are tight lower and upper bounds of $\mathbb{S}(x)$ with $\mathbb{S}(x) := \{\phi \in \mathcal{Y} | \mathfrak{l}_n(x; L^*) \leq \phi \leq \mathfrak{u}_n(x; L^*)\}$. In other words, setting parameter $L(n)$ to the best Hölder constant L^* and bounds $\underline{B} = -\infty$, $\bar{B} = +\infty$ yields a predictor $\hat{\mathfrak{f}}_\mathfrak{n}(x)$ that for every query x chooses the mid-point of the set $\mathbb{S}(x)$ of those function values that can possibly be assumed by a Hölder continuous function that interpolates the observed sample. Prediction error $\hat{\mathfrak{v}}_\mathfrak{n}(x)$ simply is the radius of the set.

For the case of $p = 1$, this approach is known as *Lipschitz interpolation* Beliakov (2006); Zabinsky et al. (2003). Since a set is utilised for interpolation, the approach is also known as *Nonlinear Set Interpolation* Milanese and Novara (2004). Specification of $\bar{B}$, $\underline{B}$ allows us to incorporate additional knowledge and constrain our set $\mathbb{S}(x)$ further. For instance, when learning friction models we might incorporate the knowledge of dealing with negative functions by setting $\bar{B} = 0$, yielding $\mathbb{S}(x) = \{\phi | \phi \leq 0\} \cap_{i=1}^{N_n} \mathbb{S}_i(x)$.

When choosing $L(n)$ to coincide with a Hölder constant of the target function, one can give strong guarantees of convergence to the target as on the tightness of the prediction bounds [e.g. Calliess (2014)]. In case a Hölder constant is unknown a priori, Milanese and Novara (2004) have proposed to utilise a proxy regression model to estimate the constant from the data. Unfortunately, as far as we can tell, all convergence guarantees and uncertainty bounds are lost in the process. On the other hand, if $L(n)$ was chosen to a very conservatively large constant then the resulting predictor tends to overfit to the noise and hence, exhibit poor prediction performance [Calliess (2016)].

As an alternative, Calliess (2014) has proposed to set

$$L(n) := \max_{i,j=1,\ldots,N_n \, \mathfrak{d}_\mathcal{X}(s_i,s_j)>0} \frac{\left\|\tilde{f}_i - \tilde{f}_j\right\|_\infty - 2\bar{\mathfrak{e}}}{\mathfrak{d}_\mathcal{X}(s_i, s_j)}$$

where $\bar{\mathfrak{e}} := \sup_x |\mathfrak{e}(x)|$ is an upper bound on the (zero-mean) noise. The resulting predictor, referred to as *LACKI* has been investigated by Calliess (2016). For cases where $\bar{\mathfrak{e}}$ is known the authors derived parameter settings for the LACKI inference rule for which the following prediction error bound was derived:

$$\mathfrak{d}_\mathcal{Y}\big(\hat{\mathfrak{f}}_\mathfrak{n}(x), f(x)\big) \in \big[0, 2\ell\, r_x^p(n) + 2\bar{\mathfrak{e}}\big]. \tag{1}$$

where $r_x(n) = \min_{s \in G_n} \mathfrak{d}_\mathcal{X}(x, s)$ is the distance of the query input x to the input data and $\ell \in \mathbb{R}, \ell \geq L^*$ is any Hölder constant of the target function f. Note, that while the prediction error *bound* still depends on knowledge of a Hölder constant, the predictor $\hat{\mathfrak{f}}_\mathfrak{n}$ itself only depends on the empirical estimator. Therefore, ℓ could be chosen conservatively large without causing the predictions to be prone to overfit to observational noise Calliess (2016).

3. DATA-BASED PREDICTION MODEL

Let the manipulable inputs of the plant at sampling time step $k \in \mathbb{N}$ be $u(k) \in \mathbb{R}^m$ and the controlled outputs be $y(k) \in \mathbb{R}^n$. It is assumed that the output signal is measured and can be used to describe the model of the system by means of the following *NARX* model of the plant

$$y(k+1) = g(x(k), u(k)) + e(k) \tag{2}$$

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 $n_x = (n_a + 1)n + n_b m$, for some memory horizon lengths $n_a, n_b \in \mathbb{N}$. The signal $e(k)$ models output noise and is assumed to be confined in a compact set $\mathcal{E}$. For notational convenience, we sometimes aggregate the inputs of g into a joint input $\xi := (x, u) \in \Xi := \mathcal{X} \times \mathcal{U}$. So, we can write $g : \Xi \to \mathcal{Y}$.

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:

$$u(k) \in \mathcal{U}, \qquad y(k) \in \mathcal{Y} \tag{3}$$

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

Assumption 1.

(1) The pseudo-metrics $\mathfrak{d}_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ and $\mathfrak{d}_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ are such that for $x = (y_1, \cdots, y_{n_a+1}, 0, \cdots, 0)$, the following condition holds
$$\mathfrak{d}_{\mathcal{X}}(x, 0) \leq \mathfrak{d}_{\mathcal{Y}}(y_1, 0) + \cdots \mathfrak{d}_{\mathcal{Y}}(y_{n_a+1}, 0).$$
(2) The function $g(\cdot)$, referred to as *ground truth function*, is Hölder continuous. That is, there exists some $L_g > 0$ and $p \in (0, 1)$ such that
$$\mathfrak{d}_{\mathcal{Y}}(g(x_1, u), g(x_2, u)) \leq L_g \mathfrak{d}_{\mathcal{X}}(x_1, x_2)^p \quad (4)$$
where $\mathfrak{d}_{\mathcal{Y}}$ stands for a pseudo-metric defined in the set of outputs $\mathcal{Y}$, and $\mathfrak{d}_{\mathcal{X}}$ stands for a pseudo-metric defined in the set of the states of the model function $\mathcal{X} = \mathcal{Y}^{(n_a+1)} \times \mathcal{U}^{nb}$. The lowest constant L_g that satisfies this condition is called the Hölder constant, L_g^*.

In this paper we assume that the prediction model will be inferred from a set of pairs of input-output experimental data
$$\mathcal{D} = \{(y(j), u(j)) \,|\, j = 1, \cdots, N_D\}.$$
This data may be extracted from some experiments or from stored historical data of the plant.

Notice that the data set can be recast as $\mathcal{D}_g = \{\big(\xi(j), y(j+1)\big) j = j_0, \cdots, N_D - 1\}$, where $j_0 = \max(n_a + 1, n_b)$. We can define $\Xi_{\mathcal{D}} := \{\big(\xi(j)\big) j = j_0, \cdots, N_D - 1\}$ and $\mathcal{Y}_{\mathcal{D}} := \{\big(y(j)\big) j = j_0, \cdots, N_D - 1\}$ as the available inputs and output samples of g available in the data.

If the required Hölder constant is not known a priori, it can be estimated from the data set $\mathcal{D}$ with a noise-robust version Calliess (2014, 2016) of Strongin's method Strongin (1973). This estimate, denoted by $L_{\mathcal{D}}$, can be utilised as a parameter in the *kinky inference* method of Calliess (2014, 2016) to obtain a predictor $\hat{g}$ of the ground-truth g, yielding the output prediction:
$$\hat{y}(k+1) = \hat{g}(x(k), u(k); L_{\mathcal{D}}). \quad (5)$$

This estimated function will be used as the prediction model in the design of a model-predictive controller that is presented in the next section. Before, this is derived however, we will conclude the present section with a brief discussion of the necessity of requiring worst-case estimation bounds of the learners in a stabilizing MPC setting.

3.1 Necessity of the worst case estimation error

In general, a nonlinear system, even in the case of no uncertainty and no constraints, might not be globally asymptotically stabilized, but only in a certain region. This is the case of the systems that are subject to constraints on the inputs and/or the states. These constraints can be inherent to the system to be controlled or to the control law to be used.

Model-predictive control requires repeated optimisation of the predicted control inputs subject to constraints. Therefore, in order to give guarantees on the controller's closed-loop performance, we need to ensure recursive feasibility. That is, we need to ensure that all constraints remain satisfiable during runtime, or equivalently, to guarantee that the controlled system will not leave the controllable region. However, since our controller will be based not on the ground-truth dynamics g, but on the learned model $\hat{g}$ inferred from a sample of the ground-truth, recursive feasibility 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.

In this paper it is assumed that a bound for the worst-case estimation error is available.

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 $\mu > 0$ with
$$\mathfrak{d}_{\mathcal{Y}}(\hat{g}(x, u; L_{\mathcal{D}}), g(x, u) + e) \leq \mu \quad (6)$$
for all $x \in \mathcal{Y}^{n_a+1} \times \mathcal{U}^{n_b}$, $u \in \mathcal{U}$ and $e \in \mathcal{E}$.

Remark 2. As explained above, if some upper bound ℓ_g of the smallest Hölder constant L_g of mapping g is available then the worst-case prediction error bound μ of the estimated model is can be given (cf. Eq. 1) as per
$$\mu = 2\ell_g R_{\mathcal{D}}^p + 2\bar{\mathfrak{e}} \quad (7)$$
where $R_{\mathcal{D}} = \sup_{\xi \in \Xi} \inf_{\xi' \in \Xi_{\mathcal{D}}} \mathfrak{d}_{\mathcal{X}}(\xi, \xi')$ is the worst-case distance of query inputs to sample inputs in the data and $\bar{\mathfrak{e}}$ is the bound of the noise, $\bar{\mathfrak{e}} = \sup_{e \in \mathcal{E}} \mathfrak{d}_{\mathcal{Y}}(e, 0)$. If the true Hölder constant is not available, then the constant could be estimated by means of a (probabilistic) validation technique from experimental data (Alamo et al. (2015)).

4. STABILIZING OFF-LINE LEARNING BASED NMPC

In this section, a model predictive controller based on prediction model learned from off-line data of the plant is derived. Since the prediction model is not accurate, the effect of the estimation error on the predictions must be analyzed to be taken into account in the design of the controller.

For this analysis, it is convenient to define the model of the plant in a state-space description as follows (Canale et al. (2014)):
$$x(k+1) = F(x(k), u(k)) + w(k) \quad (8)$$
$$y(k) = Mx(k) \quad (9)$$
where
$$F(x(k), u(k)) = (g(x(k), u(k)), y(k), \cdots, y(k - n_a + 1), u(k), \cdots, u(k - n_b + 1))$$
$$M = [I_n, 0, \cdots, 0]$$
and $w(k) = (e(k), 0, \cdots, 0)$.

Based on the set of data obtained offline $\mathcal{D}$ and the estimated Hölder constant $L_{\mathcal{D}}$, a model $\hat{g}(x, u; L_{\mathcal{D}}, \mathcal{D})$ is obtained by learning. 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
$$x(k) = (y(k), \cdots, y(k - n_a), u(k-1), \cdots, u(k - n_b)).$$
The output forecasted at sampling time $k + j$ based on the information at k is denoted as $\hat{y}(j|k)$. Given the regressive nature of the dynamic model, the predicted state is given by
$$\hat{x}(j|k) = (\hat{y}(j|k), \cdots, \hat{y}(1|k), y(k), \cdots, y(k + j - n_a), u(k + j - 1), \cdots, u(k + j - n_b))$$
This predicted state is derived from the recursion
$$\hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k), u(k+j)) \quad (10)$$
where

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

This prediction model depends on the learned model $\hat{g}$ and then depends on the provided data set $\mathcal{D}$ and the estimated Hölder constant $L_{\mathcal{D}}$, but this dependence is omitted for the sake of clarity.

Since the estimated model differs from the true dynamics of the system, there exists an error between the predicted state using the approximate model and the real state for a given sequence of future control inputs. Based on Assumption 2, the effect of the estimation error on the predictions is analyzed in the following lemma:

Lemma 3. Assume that at the 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 (10) 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 (10) with $\hat{x}(0|k+1) = x(k+1)$ and the remaining sequence of the given future control inputs.

Then the error between both predictions are such that

$$\mathfrak{d}_{\mathcal{Y}}(\hat{y}(j-1|k+1), \hat{y}(j|k)) \leq c_j \tag{11}$$

$$\mathfrak{d}_{\mathcal{X}}(\hat{x}(j-1|k+1), \hat{x}(j|k)) \leq d_j \tag{12}$$

where c_j ans d_j can be calculated from the recursion

$$c_j = L_{\mathcal{D}} d_{j-1}^p$$
$$d_{j+1} = d_j + L_{\mathcal{D}} d_j^p$$

with $c_1 = d_1 = \mu$ and $j \in \mathbb{I}_{1,N}$. ■

Remark 4. It is easy to derive that for $p = 1$, the constants c_j and d_j are directly calculated by the following formulas $d_j = (1 + L_{\mathcal{D}})^{j-1}\mu$, and $c_j = L_{\mathcal{D}}(1 + L_{\mathcal{D}})^{j-1}\mu$.

Remark 5. It can be proved that the same result holds if the error bound μ is replaced by the current value of the estimation error $\mathfrak{d}_{\mathcal{Y}}(y(k+1), \hat{y}(1|k))$.

Based on the derived bounds on the prediction error, the problem of robust constraint satisfaction is analyzed next. Firstly, the following balls are defined:

$$\mathcal{B}y(\delta) = \{y \in \mathbb{R}^n : \mathfrak{d}_{\mathcal{Y}}(y, 0) \leq \delta\}$$
$$\mathcal{B}x(\gamma) = \{x \in \mathbb{R}^{n_x} : \mathfrak{d}_{\mathcal{X}}(x, 0) \leq \gamma\}.$$

Then, the following set of tightened constraints is defined

$$\mathcal{Y}_j = \mathcal{Y} \ominus \mathcal{B}y(d_j). \tag{13}$$

This set is said to be *tightened* since $\mathcal{Y}_j \subseteq \mathcal{Y}$ and, for any output y contained in the tightened set $y \in \mathcal{Y}_j$ and for all perturbations $\Delta y \in \mathcal{B}y(d_j)$ on the output, the perturbed output is contained in the original set $\mathcal{Y}$, i.e. $y + \Delta y \in \mathcal{Y}$. As will be shown later, the set of tightened constraints will be useful to counteract the effect of the estimation error in the predictions Limon et al. (2002).

Therefore, the tightened set $\mathcal{Y}_j$ yields sufficient conditions on recursive robust constraint satisfaction (which can be utilised in the statement of the optimisation problems of the proposed controller later on):

Lemma 6. The sets $\mathcal{Y}_j$ are such that for all $y \in \mathcal{Y}_j$ and for all $\Delta y \in \mathcal{B}y(c_j)$, $y + \Delta y \in \mathcal{Y}_{j-1}$.

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 $\mathcal{Y}_{N-1}$ is non empty.

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

$$\min_{\mathbf{u}} \; V_N(x(k), \mathbf{u}) = \sum_{i=0}^{N-1} \ell(\hat{y}(i|k), u(i)) + V_f(\hat{x}(N|k))$$
$$s.t. \; \hat{x}(0|k) = x(k) \tag{14}$$
$$\hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k), u(j)), \quad j \in \mathbb{I}_{0,N-1} \tag{15}$$
$$\hat{y}(j|k) = M\hat{x}(j|k) \tag{16}$$
$$u(j) \in \mathcal{U} \tag{17}$$
$$\hat{y}(j|k) \in \mathcal{Y}_j \tag{18}$$
$$\hat{x}(N|k) \in X_f \tag{19}$$

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

Assumption 4.

(1) The stage cost function $\ell(y, u)$ is a continuous positive definite function for all $y \in \mathcal{Y}$ and $u \in \mathcal{U}$ such that $\ell(y, u) \geq \alpha_y(\|y\|)$ for a certain $\mathcal{K}$ function α_y.
(2) There exists a local control law $u = \kappa_f(x)$, a function V_f and a pair of sets $\Omega, X_f \subseteq \mathbb{R}^{n_x}$ such that:
   (a) For all $x \in \Omega$ the following conditions hold:
   $$\hat{F}(x, \kappa_f(x)) \in \Omega \ominus \mathcal{B}x(d_N)$$
   $$\kappa_f(x) \in \mathcal{U}$$
   $$Mx \in \mathcal{Y}_{N-1};$$
   (b) $X_f = \Omega \ominus \mathcal{B}x(d_N)$;
   (c) V_f is a continuous positive definite function such that
   $$\alpha_f(\|x\|) \leq V_f(x) \leq \beta_f(\|x\|)$$
   $$V_f(\hat{F}(x, \kappa_f(x))) - V_f(x) \leq -\ell(Mx, \kappa_f(x))$$
   for all $x \in X_f$.

The following theorem presents the main result of this paper:

Theorem 7. Assume that Assumptions 2, 3 and 4 hold for the optimization 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 feasible initial state $x(0)$, 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. $y(k) \in \mathcal{Y}, \forall k$.

4.1 Relaxation by means of soft constraints

Our previous results were based upon knowledge of the a priori prediction bound μ. If learning is done with the LACKI method

and bounds on the Hölder constant and observational errors are given then μ could be chosen as per (7). In practice however, cases may arise where the necessary bounds may be difficult or even impossible to know beforehand. In this case it is worth remarking that theorem 7 holds as long as the uncertainty of the closed loop trajectory is such that

$$\mathfrak{d}_{\mathcal{Y}}(y(k+1), \hat{g}(x(k), u(k), L_{\mathcal{D}})) \leq \hat{\mu}.$$

If this is not the case, as was previously discussed, the satisfaction of the hard constraints cannot be ensured in general. A sensible solution to avoid feasibility problems is to relax the constraints by considering soft constraints and design the stabilizing ingredients X_f and V_f as proposed in Assumption 4 using the estimated bound $\hat{\mu}$.

The resulting optimization problem is the following:

$$\begin{aligned}
\min_{\mathbf{u},v} \; & \sum_{i=0}^{N-1} \Big(\ell(\hat{y}(i|k), u(i)) + \alpha v_i \Big) + V_f(\hat{x}(N|k)) \\
s.t. \; & \hat{x}(0|k) = x(k) && (20)\\
& \hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k), u(j)), \quad j \in \mathbb{I}_{0,N-1} && (21)\\
& \hat{y}(j|k) = M\hat{x}(j|k) && (22)\\
& u(j) \in \mathcal{U} && (23)\\
& \hat{y}(j|k) \in \mathcal{Y}_j \oplus \mathcal{B}y(v_j) && (24)\\
& v_j \geq 0 && (25)\\
& \hat{x}(N|k) \in X_f && (26)
\end{aligned}$$

Using the results on the exact penalty functions Luenberger (1984), for a sufficiently large value of α, the derived controller guarantees the robust constraint satisfaction when possible while the constraints are relaxed when they cannot be fulfilled. The stability of the proposed controller can be derived from Theorem 7 as long as the uncertainty of the closed loop trajectory is such that $\mathfrak{d}_{\mathcal{Y}}(y(k+1), \hat{g}(x(k), u(k), L_{\mathcal{D}})) \leq \hat{\mu}$.

If this is not the case then the recursive feasibility of the terminal constraint cannot be ensured in the case that the model mismatch is larger than expected. This issue can be relaxed taking into account that the terminal constraint can be removed if the terminal cost function is weighted by an appropriate positive constant $\lambda > 0$ Limon et al. (2006); Rawlings and Mayne (2009). The new optimization control problem is the following

$$\begin{aligned}
\min_{\mathbf{u},v} \; & \sum_{i=0}^{N-1} \Big(\ell(\hat{y}(i|k), u(i)) + \alpha v_i \Big) + \lambda V_f(\hat{x}(N|k)) \\
s.t. \; & \hat{x}(0|k) = x(k) && (27)\\
& \hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k), u(j)), \quad j \in \mathbb{I}_{0,N-1} && (28)\\
& \hat{y}(j|k) = M\hat{x}(j|k) && (29)\\
& u(j) \in \mathcal{U} && (30)\\
& \hat{y}(j|k) \in \mathcal{Y}_j \oplus \mathcal{B}y(v_j) && (31)\\
& v_j \geq 0 && (32)
\end{aligned}$$

The resulting control law would be recursively feasible, but the stability of the closed-loop system would be potentially ensured only in a (potentially not small) neighborhood of the origin Limon et al. (2006).

5. STABILIZING ON-LINE LEARNING BASED NMPC

Throughout the evolution of the controlled system, new pairs of input-output data could potentially be added to the data set, yielding to an enhanced estimation of the Hölder constant. This update could be done at each sampling time or every certain number of sampling times. This is the base of the LACKI method proposed in Calliess (2016), which allows us to get estimations of the Hölder constant that are refined at each update stage, providing an enhanced estimation of the ground-truth function.

Let $\mathcal{D}(k)$ denote the set of all data compiled until the sampling time k and let $L(k)$ be the estimated Hölder constant for that set provided by the LACKI algorithm. Similarly to Aswani et al. (2013), we will use two different prediction models:

- The initial or off-line learning model:
$$\hat{x}(k+1) = \hat{F}(x(k), u(k), L_0, \mathcal{D}_0)$$
This model is derived off-line from the historical data $\mathcal{D}_0 := \mathcal{D}(0)$ available before closing the loop (i.e. at time $k = 0$) and it is assumed that a guaranteed bound on the estimation error is available. The prior model is utilised to provide guaranteed estimations of the real plant and it will be used to ensure robust constraint satisfaction.
- The updated or on-line learning model:
$$\hat{x}(k+1) = \hat{F}(x(k), u(k), L(k), \mathcal{D}(k))$$
This model may provide a probably more accurate estimation of the real plant, but no guarantee of the estimation error is given. Subsequently, this model will be used to enhance the closed-loop behavior of the controlled plant using the updated predictions to derive the control law.

Based on these prediction models, the following optimization problem $P_N^u(x; L_0, \mathcal{D}_0, L(k), \mathcal{D}(k))$ is proposed:

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{L}} \; & \sum_{i=0}^{N-1} \ell(\hat{y}_u(i|k), u(i)) + V_f(\hat{x}_u(N|k)) \\
s.t. \; & \hat{x}_u(0|k) = x(k) \\
& \hat{x}_u(j+1|k) = \hat{F}(\hat{x}_u(j|k), u(j), L(k), \mathcal{D}(k)),\ j \in \mathbb{I}_{0,N-1} \\
& \hat{y}_u(j|k) = M\hat{x}_u(j|k) \\
\\
& \hat{x}_s(j+1|k) = \hat{F}(\hat{x}_s(j|k), u(j), L_0, \mathcal{D}_0),\ j \in \mathbb{I}_{0,N-1} \\
& \hat{y}_s(j|k) = M\hat{x}_s(j|k) \\
& u(j) \in \mathcal{U} \\
& \hat{y}_s(j|k) \in \mathcal{Y}_j \\
& \hat{x}_s(N|k) \in X_f
\end{aligned}$$

In order to derive the robust stability of the proposed controller, the following assumption on the estimation method is required

Assumption 5. There exists a certain time instant k^* such that

$$\hat{g}(x, u, L(k), \mathcal{D}(k)) \in \hat{g}(x, u, L_0, \mathcal{D}_0) \oplus B(\mu)$$

or equivalently,

$$\mathfrak{d}_{\mathcal{Y}}(\hat{g}(x, u, L(k), \mathcal{D}(k)), \hat{g}(x, u, L_0, \mathcal{D}_0)) \leq \mu$$

for all x,u.

Notice that from this assumption we have that

$$\mathfrak{d}_{\mathcal{X}}(\hat{F}(x, u, L(k), \mathcal{D}(k)), \hat{F}(x, u, L_0, \mathcal{D}_0)) \leq \mu$$

It can be proved that the LACKI method satisfies this assumption.

Based on this assumption, the following lemma can be derived:

Lemma 8. Assume that Assumption 5 holds and that at the sampling time $k \geq k^*$ and consider that the state of the plant is $x(k)$ and the sequence of future control inputs is known. Let $\hat{x}_s(j|k)$ and $\hat{y}_s(j|k)$ be the predicted states and outputs respectively corresponding to the given sequence of future control inputs using the guaranteed prior model.
Assume that at sampling time $k+1$, the states and the outputs of the plant are predicted for the remaining sequence of the given future control inputs taking into account the measured output $y(k+1)$. Let $\hat{x}_s(j|k+1)$ and $\hat{y}_s(j|k+1)$ be the sequences of states and outputs predicted with the prior model and let $\hat{x}_u(j|k+1)$ and $\hat{y}_u(j|k+1)$ be the predicted sequences using the updated model at $k+1$.

Then the errors between both predictions are such that

$$\mathfrak{d}_{\mathcal{X}}(\hat{x}_u(j-1|k+1), \hat{x}_s(j|k)) \leq d_j \tag{33}$$

where d_j is as defined in Lemma 3.

Based on this lemma, we are ready to state the closed-loop stability of the proposed controller

Theorem 9. Consider that Assumptions 3, 4 and 5 hold. Let $\kappa_N^u(x)$ be the control law derived from the solution of $P_N^u(x)$ applied using a receding horizon policy. Then for all feasible $x(0)$, the system controlled by the control law $u(k) = \kappa_N^u(x(k))$ is input-to-state stable w.r.t to the estimation error of the LACKI model and the constraints are satisfied along the time, i.e. $y(k) \in \mathcal{Y}$.

6. AN ILLUSTRATION

Most of the theoretical derivations thus far were concerned with robust MPC design in a manner such that the controller and its guarantees can be connected to learning methods such as LACKI. When using such a learning method for identifying the dynamics of the underlying plant, we expect our adaptive controller to improve its performance the more data becomes available. To illustrate this behaviour, as well as the viability of our approach in general, we will now attend to a simple example:

We simulated the discrete-time version of a torque-actuated single-pendulum with continuous-time dynamics $\dot{x} = a(x) + b(x)u$ where $a(x) := -\frac{g}{l}\sin(x_1) - \frac{r}{ml^2}x_2$ and $b(x) = \frac{1}{ml^2}$. Here, $x_1 = q, x_2 = \dot{q} \in \mathbb{R}$ are joint angle position and velocity, $r = 0.01$ denotes a friction coefficient, $g = 9.81$ is acceleration due to gravity $l = 1$ is the length and $m = 0.1$ the mass of the pendulum.

The control input $u \in [-.5, .5]$ applies a torque to the joint that corresponds to joint-angle acceleration. The pendulum could be controlled by application of a torque u to its pivotal point. $q = 0$ encodes the pendulum pointing downward and $q = \pi$ denotes the position in which the pendulum is upward. Given an initial configuration $x_0 = [0, 0]$ we desired to steer the state to a terminal configuration $\xi = [\pi, 0]$. The bounds on the control inputs were set so that the controller cannot lift up the pendulum directly to the goal position but has to plan ahead to perform a swing up. To convert the system to a discrete-time system, we chose a first-order Euler approximation with sample time $\Delta = 0.1[sec.]$.

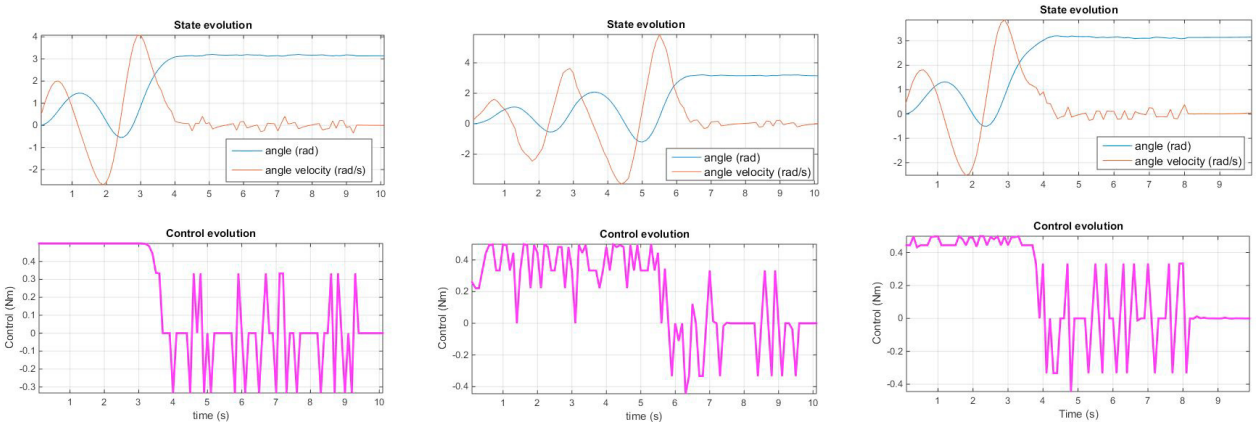
As a first test of the viability of nonlinear MPC in this setting, we designed an MPC controller as per Eq. 27-32 but where the dynamics $\hat{F}$ were set to coincide with the true dynamics. Furthermore, we chose a prediction horizon of $N = 5$ time steps and, and tuned the parameters of V and λ via linearisation around the equilibrium point. To solve the nonlinear optimisation problem in each step of the receding horizon control process, we employed the DIRECT global optimisation method [Jones et al. (1991)] with a computational budget of 2000 function evaluations. The resulting trajectory is depicted in Fig. 1(a). As can be seen from the plots, the controller was capable of performing the swing up and stabilising the pendulum in the up-right position (up to some jitter). We explain the remaining jitter to be an artefact of the optimisation method failing to find the global optima within the given budget.

As a first test of our learning-based MPC, we next modified the MPC, substituting the true dynamics with a LACKI predictor $\hat{\mathfrak{f}}_{\mathfrak{n}}$ trained on a data set on an even-spaced grid such that that training inputs had a distance of $R_{\mathcal{D}_n} = \max_{i,j} \|s_i - s_j\|_\infty = 0.6$. We then repeated the simulation with this learning-based conroller, LACKI-MPC, controlling the torques. The resulting trajectories are depicted in Fig.1(b). As can be seen, the controller again managed to perform the swing-up and stabilisation. However, this time, most-likely due to the model inaccuracies of the trained LACKI predictor, the pendulum had to swing-back and forth repeatedly until the swing up succeeded.

To illustrate the benefit of additional data, we once more repeated the experiment in the same setup but where LACKI was given more training data. In this second run of LACKI-MPC, the LACKI learner was trained on a finer grid with $R_{\mathcal{D}_n} = 0.4$. As can be seen in Fig. 1(c), the controller managed to achieve the swing-up quicker. And, the closed-loop behaviour very closely mimicked the characteristics of those of the plant controlled by the MPC with access to the true dynamics (as per Fig. 1(a)).

7. CONCLUSIONS

In this paper, a new learning-based MPC control approach was proposed. Here control actions were based on a sequence of nonlinear MPC optimisation problems whose prediction models were learned on the basis of input-output observations of the underlying plant. Several assumptions were discussed under which input-to-state stability of the resulting closed-loop dynamics can be proven. In contrast to some of the earlier work going in the same direction as ours, our analysis also gives guarantees on recursive feasibility of unrestricted (but typically continuous) nonlinear dynamics without having to make model assumptions that in effect limit the constraints to be convex. A crucial requirement for our guarantees to hold was that worst-case bounds around the predictions of the trained learning models could be given. Providing an example of such a learning method, we considered to employ LACKI to this end [Calliess (2014, 2016)], a recently developed extension to NSM and Lipschitz interpolation methods [Sukharev (1978); Milanese and Novara (2004)] that had previously been proposed to be utilised in learning-based MPC [Canale et al. (2014)]. To illustrate the viability of our ideas, simulations of a torque actuated pendulum were provided. Here the learning-based MPC proved capable of learning how to swing up and stabilise in the presence of sufficient learning experience.



(a) MPC based on true model. (b) LACKI-MPC with some data. (c) LACKI-MPC with more data.

Figure 1. Simulations of a single pendulum. Left: Controlled by a nonlinear MPC knowing the true nonlinear dynamics. Centre: The trajectories resulting from utilising our MPC controller with soft constraints based on a coarse LACKI model trained on offline data. Right: LACKI-MPC based on a refined model.

7.1 Future work.

While the theoretical results are encouraging, a lot of avenues of further investigation remain open. For instance, we have found that in our simulations, especially in higher-dimensional systems, the performance of the controller appeared to be sensitive to the computational budget and chosen nonlinear optimisation algorithm. This is especially true since the NSM and LACKI methods are not smooth. Future work will investigate alternative optimisation algorithms that are more suitable for such function classes. From the other end, we will investigate the utilisation of smooth regression methods such as neuronal networks or Gaussian processes [Rasmussen and Williams (2006); Kocijan and Murray-Smith (2005)] within the learning-based MPC. While we already have conducted first experiments in that regard that are encouraging, we yet have to investigate in how far these learning approaches can be proven to meet the assumptions required to connect to our theory. We will also study more carefully the on-line methods as well as techniques to reduce the conservativeness of the proposed approaches. Other directions are to investigate the addition of learning-oriented cost terms into the MPC objective function [e.g. similar in spirit to Alpcan (2011)] that facilitate exploration or explicitly encourage exploitation, giving preference to trajectories that traverse better-explored regions of state-control space over more uncertain ones.

REFERENCES

Alamo, T., Tempo, R., Luque, A., and Ramirez, D.R. (2015). Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. *Automatica*, 52, 160–172.

Alpcan, T. (2011). Dual control with active learning using Gaussian process regression. *Arxiv preprint arXiv:1105.2211*.

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*.

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.

Jones, D., Peritunen, C., and Stuckman, B. (1991). Lipschitzian optimization without the Lipschitz constant. *JOTA*, 79(1).

Kocijan, J. and Murray-Smith, R. (2005). Nonlinear Predictive Control with a Gaussian. *Lecture Notes in Computer Science 3355, Springer*, 185–200.

Limon, D., Alamo, T., and Camacho, E. (2002). Input-to-state stable mpc for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 4, 4619–4624. IEEE.

Limon, D., Alamo, T., Salas, F., and Camacho, E.F. (2006). On the stability of MPC without terminal constraint. 42, 832–836.

Limon, D., Callies, J., and Maciejowski, J. (2017). Stabilizing design of data-based nonlinear MPC. preliminary results. Technical report, ArXiv pre-print.

Luenberger, D.E. (1984). *Linear and Nonlinear Programming*. Addison-Wesley.

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

Rasmussen, C. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob-Hill Publishing, 1st edition.

Strongin, R.G. (1973). On the convergence of an algorithm for finding a global extremum. *Engineering in Cybernetics*.

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*.