2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Data-Driven Predictive Control for Nonlinear Systems Using Feature Selection

Dirk Reinhardt, Sébastien Gros

***Abstract*—This paper proposes a model-free predictive control method for nonlinear systems based on a data-driven representation of the system behavior and a predictive control formulation similar to other designs using Willems' Fundamental Lemma. To select a subset of available trajectories for representing the system behavior, we use a feature selection algorithm inspired by Lasso regression. This algorithm prevents interferences between control objectives and predictions and is well-suited for nonlinear systems. The construction is supported by a simple theoretical analysis, and simulation-based experiments demonstrate that the proposed approach can control nonlinear systems.**

***Index Terms*—Data-driven Control, Model-free Control, Predictive Control, Feature Selection**

I. Introduction

A. Background and Motivation

Model Predictive Control (MPC) is a widely used technique in control systems, where a model is employed to predict the future behavior of a given system and optimize the control input sequence, satisfying constraints while minimizing a cost function [1]. A major challenge in deploying MPC is finding an accurate model of the system, in particular for nonlinear and stochastic systems. An inaccurate model may lead to inaccurate predictions, especially over long time horizons, and result in poor control solutions.

To address this challenge, data-driven approaches have been proposed that do not require a state-space representation or a one-step ahead model of the system dynamics. One such approach is Subspace Predictive Control (SPC) [2], which uses a linear multi-step predictor obtained by projecting the data onto a low-dimensional subspace. SPC requires only past input-output trajectories and works best for LTI systems. Other approaches to data-driven predictive control, e.g. [3], [4], lean on the Fundamental Lemma introduced by Willems et al. [5], which states that the future output of a system can be expressed as a linear combination of past input-output trajectories.

In particular after the introduction of DeePC [6], which blends control objective and system identification in a single Optimal Control Problem (OCP), the community found renewed interest in data-driven predictive control, with recent theoretical discussions addressing closed-loop robustness [7], application to nonlinear systems [8] or noise mitigation [9]. Different contributions recently established the equivalence between DeePC and SPC [9], [10], and compared DeePC-like approaches to classic system identification and control [11], [12], possibly using linear regression or subspace identification [13]. The necessary predictor updates in closed-loop control are an open problem [14].

A core ingredient of DeePC is the regularization of the linear combination of past trajectories. A prominent choice in this context is the ℓ_1-norm or Least Absolute Shrinkage and Selection Operator (LASSO) [15] on the weights of the linear combination. This promotes sparsity of these weights, resulting in the selection of a small subset of trajectories to construct the prediction, referred to as *feature selection* in the machine-learning community. However, this subset selection is not guaranteed to produce good predictions, as we discuss in more detail in Section II. The cross-effect between identification and control, intrinsic to DeePC, is another issue [11] and a proper tuning of the regularization can be challenging or even impossible in practice [16]. Other types of regularization terms such as ℓ_2-norm or hybrid formulations are not free of these issues [12]. Finally, the computational demand of the resulting control scheme increases dramatically with the size of the data set, which is a general problem for data-driven approaches. Given that nonlinear systems require increasing amounts of data to capture their behavior, alternative approaches are necessary.

B. Contribution and Outline

In this paper, we present a modified version of DeePC that is specifically designed to tackle nonlinear problems in a more efficient manner. Our approach involves two simple yet crucial changes that make a significant impact. Firstly, we pre-select past trajectories based on their proximity to the current system trajectory and optimality with respect to the chosen control objective. We employ a convex heuristic for subset selection to ensure that the control scheme operates on a local representation of the system behavior. This process can be seen as a feature selection, as well as an implicit linearization, enabling more meaningful predictions. Secondly, we ensure that the prediction is an interpolation from the data rather than an extrapolation, thereby guaranteeing the validity of the local approximation constructed from the pre-selection process.

This research was funded by the Research Council of Norway (RCN) through the project Safe Reinforcement Learning using MPC (SARLEM), grant number 300172. Dirk Reinhardt and Sébastien Gros are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway (e-mail: {dirk.p.reinhardt, sebastien.gros}@ntnu.no)

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

One of the advantages of our approach is that the computational demand of the resulting control scheme is limited by the size of the pre-selected set. Although the pre-selection process has its own computational demand, it is limited to evaluating basic functions on the data, such as norms and stage costs, as well as sorting procedures. We show that the prediction error is dominated by the distance between the system's current input-output data and the input-output data in a dictionary used to form predictions. Another important factor is the curvature of the map from past input-output data and future input data to the system's future output data, i.e. the nonlinearity of the system dynamics. The simulation-based experiment shows that the proposed approach produces accurate predictions for a nonlinear system which is not possible with the classic DeePC approach.

The remainder of the paper is organized as follows. In Section II, we provide a brief overview of the DeePC approach and discuss its limitations. In Section III, we qualitatively analyze the effect of the nonlinearity of the system on the prediction error and provide a solution idea on how to reduce it. We present our approach in Section IV and discuss the details of the pre-selection process. Section V presents the results of a simulation-based experiment, and we conclude the paper in Section VI.

II. Background

For an LTI system with inputs $\mathbf{u} \in \mathbb{R}^m$ and output $\mathbf{y} \in \mathbb{R}^p$, at any time k, the future output trajectories of the system, i.e. $\mathbf{y}_k^{\mathrm{f}} = \begin{bmatrix}\mathbf{y}_k^\top & \dots & \mathbf{y}_{k+N}^\top\end{bmatrix}^\top$ are given by

$$\mathbf{y}_k^{\mathrm{f}} = \varphi\left(\mathbf{u}_k^{\mathrm{f}}, \mathbf{u}_k^{\mathrm{p}}, \mathbf{y}_k^{\mathrm{p}}\right), \tag{1}$$

where $\mathbf{y}_k^{\mathrm{f}}$ are the N future outputs, $\mathbf{y}_k^{\mathrm{p}}$, $\mathbf{u}_k^{\mathrm{p}}$ denote recent past output and input data, and

$$\mathbf{u}_k^{\mathrm{f}} = \begin{bmatrix}\mathbf{u}_k^\top & \dots & \mathbf{u}_{k+N}^\top\end{bmatrix}^\top \tag{2}$$

denotes the future inputs to the system. The function φ is linear, and requires the past input-output data to be of sufficient length L to describe the future output of the system, i.e.

$$\mathbf{y}_k^{\mathrm{p}} = \begin{bmatrix}\mathbf{y}_{k-L}^\top & \dots & \mathbf{y}_{k-1}^\top\end{bmatrix}^\top \tag{3a}$$

$$\mathbf{u}_k^{\mathrm{p}} = \begin{bmatrix}\mathbf{u}_{k-L}^\top & \dots & \mathbf{u}_{k-1}^\top\end{bmatrix}^\top. \tag{3b}$$

The linearity of φ in (1) naturally entails that $\mathbf{y}_k^{\mathrm{f}}$, $\mathbf{u}_k^{\mathrm{f}}$, $\mathbf{y}_k^{\mathrm{p}}$, $\mathbf{u}_k^{\mathrm{p}}$ are jointly restricted to a specific subspace that includes all input-output trajectories of the system. Consequently, there is a vector $\boldsymbol{\alpha}$ with elements α_i such that:

$$\begin{bmatrix}\mathbf{u}_k^{\mathrm{p}}\\ \mathbf{y}_k^{\mathrm{p}}\\ \mathbf{u}_k^{\mathrm{f}}\\ \mathbf{y}_k^{\mathrm{f}}\end{bmatrix} = \sum_{i\in\mathcal{D}} \alpha_i \begin{bmatrix}\mathbf{u}_i^{\mathrm{p}}\\ \mathbf{y}_i^{\mathrm{p}}\\ \mathbf{u}_i^{\mathrm{f}}\\ \mathbf{y}_i^{\mathrm{f}}\end{bmatrix} \tag{4}$$

where $\mathcal{D}$ is a set of (time) indices of past trajectories. These past trajectories ought to be selected such that they span the space of all possible trajectories of the system (subject to specified L, N). In particular, [5] showed that it is sufficient to select these past trajectories such that they form the columns of a Hankel matrix. A rank condition on that matrix ensures that the set of past trajectories is sufficient to span the full trajectory space. If the set of past trajectories is not sufficient to span the full trajectory space, the solutions are restricted to the subspace spanned by this limited set of trajectories, which means that the solution in general does not match the full system behavior.

DeePC approaches commonly formulate predictive control schemes based on (4) for given $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$ as

$$\min_{\boldsymbol{\alpha},\, \mathbf{u}_k^{\mathrm{f}},\, \mathbf{y}_k^{\mathrm{f}}} \quad J_{\mathrm{ctrl}}(\mathbf{u}_k^{\mathrm{f}}, \mathbf{y}_k^{\mathrm{f}}) + \lambda \cdot J_{\mathrm{id}}(\boldsymbol{\alpha}) \tag{5a}$$

$$\text{s.t.} \quad (4), \quad \mathbf{y}_k^{\mathrm{f}} \in \mathcal{Y}, \quad \mathbf{u}_k^{\mathrm{f}} \in \mathcal{U}, \tag{5b}$$

where $J_{\mathrm{ctrl}}(\mathbf{u}, \mathbf{y})$ defines the control objective of the predictive control problem, and $J_{\mathrm{id}}(\boldsymbol{\alpha})$ is a regularization term that is required for practical purposes, i.e. to mitigate the effect of noise and to make the solution $\boldsymbol{\alpha}$ unique. Different J_{id} terms have been proposed in the literature, e.g., $J_{\mathrm{id}}(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_1$ in [6], $J_{\mathrm{id}}(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_2^2$ in [11], or hybrid formulations [11], possibly including orthogonal projections.

One ought to observe that, if omitting J_{ctrl} from (5), the predictions obtained from (5) using $J_{\mathrm{id}}(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_2^2$ are equivalent to predictions obtained via a linear least squares regressor that minimizes the prediction errors on the training data. Using $J_{\mathrm{id}}(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_1$ instead, the predictions are equal to a linear regressor estimated using least squares, but with the additional feature that only a small subset of the training data is used for the estimation. Both cases are discussed in [12]. In either case, it is questionable whether the predictions are better than those obtained by Subspace Identification [13] or by a linear least squares regressor. Performing the regression outside the predictive control scheme is typically computationally more efficient than the mixed objective formulation (5), hence allowing for larger datasets to be used.

The ℓ_1-norm regularization may help in the nonlinear case to favor data that is close to the current state of the system, and some empirical examples support this claim. However, it is straightforward to construct simple examples where the ℓ_1-norm regularization does not select such data, see Fig. Fig. 1 for a sketch of this issue.

Independent of the specific choice of J_{id}, it is an open debate in the community whether the mixed objective formulation (5) is a benefit or a source of difficulties. Some argue for the former via stating that the mixed objective formulation amounts to control-tailored system identification, though a formalization of that statement is arguably lacking. Counterarguments are based on the observation that – in the presence of noise, disturbances or nonlinearities – the mixed objective can corrupt the original control objective and renders the resulting input-output sequences either suboptimal (corrupted control sequence for better prediction accuracy) or inaccurate (corrupted prediction sequence for better control performance). The compromise between control and prediction performance is typically achieved by tuning the weight λ in (5). Examples for such a trade-off for varying λ are given in [11], showing that the effect of inaccurate predictions for low λ and the effect of suboptimal control for high λ can be severe, i.e. in the range of 8000%-14000% error with respect to the baseline optimal solution for the original control problem. Even in the optimal band of λ values, the relative error is close to 1000% and hence the mixed objective is an issue for all regularizations.

A. Regularization via exact penalties

Among the possible choices of regularization J_{id} in (5), regularizations based on exact penalties, such as the ℓ_1-norm deserve a special attention. Indeed, exact penalties on the vector $\boldsymbol{\alpha}$ will promote sparsity in the solution $\boldsymbol{\alpha}$ produced by solving (5), see e.g. [17]. In particular, the choice of the LASSO, i.e. ℓ_1-regularization, is typically described as a convex relaxation of the subset selection regression problem [15]. In that context, (5) with J_{id} based on the ℓ_1-norm can be construed as selecting the least amount of past trajectories necessary to satisfy (4). The Machine Learning or statistical learning communities regard that as a *feature selection*, where the features are past trajectories to be used to explain future ones.

It has been observed in some simulation-based studies, see e.g. [6], [11], that using the ℓ_1-norm for J_{id} in (5) can deliver good results when applied to nonlinear systems. In this paper, we aim at building an understanding why this observation can sometimes hold, and at improving the performance of this approach by performing a genuine feature selection rather than using an indirect, heuristic approach based on the ℓ_1-norm.

III. Feature selection for nonlinear problems

We ought to observe first that relationship (1) is not necessarily limited to linear systems. Indeed, the observability of the system state from the input-output sequences $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$ is a sufficient condition for (1) to hold. However, relationship (4) is tied to the linearity of function φ and therefore does not apply in general for nonlinear systems. Nonetheless, (4) can capture φ locally even for nonlinear systems, provided that it obeys some basic properties. In order to make these statements more specific, let us consider $\mathbf{u}_k^{\mathrm{f}}$, $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$ provided as linear combinations of the data, i.e.

$$\underbrace{\begin{bmatrix}\mathbf{u}_k^{\mathrm{p}}\\ \mathbf{y}_k^{\mathrm{p}}\\ \mathbf{u}_k^{\mathrm{f}}\end{bmatrix}}_{:=\mathbf{p}_k} = \sum_{i\in\mathcal{D}} \boldsymbol{\alpha}_i \underbrace{\begin{bmatrix}\mathbf{u}_i^{\mathrm{p}}\\ \mathbf{y}_i^{\mathrm{p}}\\ \mathbf{u}_i^{\mathrm{f}}\end{bmatrix}}_{:=\mathbf{d}_i} \tag{6a}$$

$$\mathbf{y}_k^{\mathrm{f}} = \varphi\left(\mathbf{u}_k^{\mathrm{f}}, \mathbf{u}_k^{\mathrm{p}}, \mathbf{y}_k^{\mathrm{p}}\right) := \phi\left(\mathbf{p}_k\right) = \phi\left(\sum_{i\in\mathcal{D}} \alpha_i \mathbf{d}_i\right), \tag{6b}$$

where we define ϕ as a short notation for φ with $\mathbf{u}_k^{\mathrm{f}}, \mathbf{u}_k^{\mathrm{p}}, \mathbf{y}_k^{\mathrm{p}}$ stacked together as a single argument. When φ and hence ϕ are linear, (4) becomes a consequence of (6) as ϕ trivially satisfies the homogeneity property

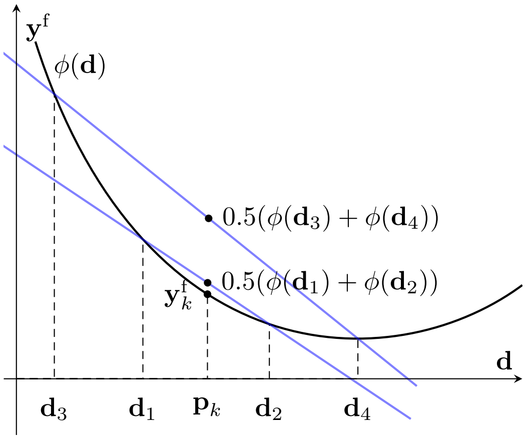
$\mathbf{y}^{\mathrm{f}}$
$\phi(\mathbf{d})$
$0.5(\phi(\mathbf{d}_3)+\phi(\mathbf{d}_4))$
$0.5(\phi(\mathbf{d}_1)+\phi(\mathbf{d}_2))$
$\mathbf{y}_k^{\mathrm{f}}$
$\mathbf{d}$
$\mathbf{d}_3$
$\mathbf{d}_1$
$\mathbf{p}_k$
$\mathbf{d}_2$
$\mathbf{d}_4$

Fig. 1. Sketch of the problem of predicting the output of a nonlinear system from its recent past input/output data using LASSO regression. The recent data and future control input sequence are collected in the vector $\mathbf{p}_k$ which produces the future output $\mathbf{y}_k^{\mathrm{f}}$ through the nonlinear function ϕ. The LASSO regression problem is to find the coefficients α_i that minimize the sum of the absolute values of the coefficients, i.e. the ℓ_1 norm. For the given example, $\mathbf{p}_k = \sum_i \alpha_i \mathbf{d}$ holds for both $\boldsymbol{\alpha} = [0.5,\ 0.5,\ 0,\ 0]$ and $\boldsymbol{\alpha} = [0,\ 0,\ 0.5,\ 0.5]$, with equal ℓ_1 norm. However, $\mathbf{d}_1$, $\mathbf{d}_2$ are closer to $\mathbf{p}_k$ than $\mathbf{d}_3$, $\mathbf{d}_4$ are, hence the ℓ_1 norm does not reflect the proximity of the data to $\mathbf{p}_k$. The similarity to the current trajectory in a behavioral sense propagates to the predicted output. Consequently, the prediction based on $0.5(\phi(\mathbf{d}_1)+\phi(\mathbf{d}_2))$ can be expected to be more accurate than the prediction based on $0.5(\phi(\mathbf{d}_3)+\phi(\mathbf{d}_4))$. We propose to pre-select the trajectories $\mathbf{d}_i$ that are close to $\mathbf{p}_k$ and to use them to form the prediction.

$$\phi\left(\sum_{i\in\mathcal{D}} \alpha_i \mathbf{d}_i\right) = \sum_{i\in\mathcal{D}} \alpha_i \phi\left(\mathbf{d}_i\right) \tag{7}$$

which allows for the use of (4) to form exact predictions in the absence of stochasticity. However, when φ and ϕ are nonlinear, (7) does not hold in general. In this case, the prediction accuracy of (4) may be very poor for a wide domain of input-output trajectories. It is possible to show that (4) can still be used to predict the output of a nonlinear system, provided that the library of trajectories $\mathcal{D}$ is sufficiently close to the recent history of past input/output data $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$ of the system. The following proposition provides qualitative conditions for (4) to be an accurate predictor, and establishes a bound on the prediction error when using it for nonlinear systems. Note that we aim at providing a conceptual analysis supporting the proposed construction, and not necessarily a methodology to be used in practice.

Proposition 1: Let us consider construction (4) based on the trajectories of a nonlinear system obeying (1), i.e.

$$\begin{bmatrix}\mathbf{u}_k^{\mathrm{p}}\\ \mathbf{y}_k^{\mathrm{p}}\\ \mathbf{u}_k^{\mathrm{f}}\\ \hat{\mathbf{y}}_k^{\mathrm{f}}\end{bmatrix} = \sum_{i\in\mathcal{D}} \alpha_i \begin{bmatrix}\mathbf{u}_i^{\mathrm{p}}\\ \mathbf{y}_i^{\mathrm{p}}\\ \mathbf{u}_i^{\mathrm{f}}\\ \mathbf{y}_i^{\mathrm{f}}\end{bmatrix} \tag{8}$$

such that the output prediction is provided by:

$$\hat{\mathbf{y}}_k^{\mathrm{f}} = \sum_{i \in \mathcal{D}} \alpha_i \phi\left(\mathbf{d}_i\right) \tag{9}$$

with $\boldsymbol{\alpha}$ satisfying (6a). Let us further assume that conditions

$$\sum_{i \in \mathcal{D}} \alpha_i = 1, \qquad \alpha_i \geq 0 \tag{10}$$

hold on $\boldsymbol{\alpha}$, and that ϕ is at least twice continuously differentiable. Then the error between the prediction $\hat{\mathbf{y}}_k^{\mathrm{f}}$ and the true trajectory $\mathbf{y}_k^{\mathrm{f}}$, i.e.

$$e_\phi(\mathcal{D}, \boldsymbol{\alpha}) = \mathbf{y}_k^{\mathrm{f}} - \hat{\mathbf{y}}_k^{\mathrm{f}}, \tag{11}$$

is bounded by

$$\|e_\phi(\mathcal{D}, \boldsymbol{\alpha})\| \leq \max_{\mathbf{v} \in C_{\mathcal{D}}(\boldsymbol{\alpha})} \left\|\nabla^2 \phi\left(\mathbf{v}\right)\right\| V_{\mathcal{D}}\left(\boldsymbol{\alpha}\right). \tag{12}$$

In (12), $C_{\mathcal{D}}\left(\boldsymbol{\alpha}\right)$ is the convex hull of the past trajectories $\mathbf{d}_i$ with associate $\alpha_i > 0$, and

$$V_{\mathcal{D}}\left(\boldsymbol{\alpha}\right) = \max_{\mathbf{v}, \mathbf{w} \in C_{\mathcal{D}}(\boldsymbol{\alpha})} \|\mathbf{v} - \mathbf{w}\|^2$$

is an indicator of the size of $C_{\mathcal{D}}$, and therefore of how close the selected past trajectories $\mathbf{d}_i$ are to each other.

Proof: Consider a quadratic approximation of ϕ centered at $\mathbf{p}_k$:

$$\phi\left(\mathbf{d}_i\right) \approx \phi\left(\mathbf{p}_k\right) + \nabla\phi\left(\mathbf{p}_k\right)\left(\mathbf{d}_i - \mathbf{p}_k\right) + \tfrac{1}{2}\nabla^2\phi(\mathbf{p}_k)\left(\mathbf{d}_i - \mathbf{p}_k\right)^2. \tag{13}$$

Using (13) and (9) for output predictions results in

$$\begin{aligned} \hat{\mathbf{y}}_k^{\mathrm{f}} =& \sum_{i \in \mathcal{D}} \alpha_i \phi\left(\mathbf{p}_k\right) + \sum_{i \in \mathcal{D}} \alpha_i \nabla\phi\left(\mathbf{p}_k\right)\left(\mathbf{d}_i - \mathbf{p}_k\right) \\ &+ \frac{1}{2}\sum_{i \in \mathcal{D}} \alpha_i \nabla^2\phi(\mathbf{p}_k)\left(\mathbf{d}_i - \mathbf{p}_k\right)^2. \end{aligned} \tag{14}$$

Under conditions (6a) and (10) equation (14) can be rewritten as

$$\mathbf{e}_\phi(\mathcal{D}, \alpha) := \phi\left(\mathbf{p}_k\right) - \hat{\mathbf{y}}_k^{\mathrm{f}} = -\frac{1}{2}\sum_{i \in \mathcal{D}} \alpha_i \nabla^2\phi(\mathbf{p}_k)\left(\mathbf{d}_i - \mathbf{p}_k\right)^2. \tag{15}$$

Because of conditions (10) and $\mathbf{p}_k \in C_{\mathcal{D}}\left(\boldsymbol{\alpha}\right)$, the error term is bounded by

$$\begin{aligned} \|\mathbf{e}_\phi(\mathcal{D}, \boldsymbol{\alpha})\| &= \left\| -\frac{1}{2}\sum_{i \in \mathcal{D}} \alpha_i \nabla^2\phi(\mathbf{p}_k)\left(\mathbf{d}_i - \mathbf{p}_k\right)^2 \right\| && (16) \\ &\leq \max_{\mathbf{v}, \mathbf{w} \in C_{\mathcal{D}}(\boldsymbol{\alpha})} \left\|\nabla^2\phi\left(\mathbf{v}\right)\left(\mathbf{v} - \mathbf{w}\right)^2\right\| && (17) \\ &\leq \max_{\mathbf{v} \in C_{\mathcal{D}}(\boldsymbol{\alpha})} \left\|\nabla^2\phi\left(\mathbf{v}\right)\right\| V_{\mathcal{D}}(\boldsymbol{\alpha}) && (18) \end{aligned}$$

■

A. Discussion of Proposition 1

While fairly straightforward, Proposition 1 points to critical aspects for predictions in the form (4) to work effectively for nonlinear systems. More specifically:

1) The past trajectories used as data in linear combination (4) ought to be in proximity to the prediction delivered by (4), such that $V_{\mathcal{D}}$ is small. Note that *proximity* here is in the space spanned by the past input and outputs and the future inputs, i.e. $\mathbf{u}_i^{\mathrm{p}}$, $\mathbf{y}_i^{\mathrm{p}}$, $\mathbf{u}_i^{\mathrm{f}}$. This suggests that the $\boldsymbol{\alpha}$ used in the linear combination (4) ought to be non-zero only for trajectories $\mathbf{u}_i^{\mathrm{p}}$, $\mathbf{y}_i^{\mathrm{p}}$, $\mathbf{u}_i^{\mathrm{f}}$ that are close to the predicted one, i.e. $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$, $\mathbf{u}_k^{\mathrm{f}}$. Note that while $V_{\mathcal{D}}$ can be made small by choosing trajectories in $\mathcal{D}$ that are close to each other, they also need to be close to the recent history of the system to make sure the prediction starts from similar initial conditions.
2) Conditions (10) ought to hold on $\boldsymbol{\alpha}$, such that the linear combination (4) seeks predictions by *interpolating* from previous trajectories rather than *extrapolating* from them. Indeed, an interpolation between neighboring trajectories will remain in that neighborhood; however, extrapolations can, in principle, leave that neighborhood, thus losing locality.
3) The nonlinear dynamics φ and their short form ϕ have an impact on the accuracy of the prediction via the Hessian $\left\|\nabla^2\phi\right\|$. Non-differentiable dynamics are precluded from this construction, and the more strongly nonlinear the system is (large Hessian), the smaller $V_{\mathcal{D}}$ should be. Similar arguments requiring sufficiently slow nonlinear dynamics when employing linear tracking MPC have been made in [8].

One can in addition observe that inequality (12) is possibly not tight, as it is based on bounds using the norm of the Hessian. Tighter bounds can arguably be developed, but not presented here for the sake of brevity.

These observations point to important caveats in using a predictive control approach (5) with a conventional exact convex penalty such as the ℓ_1-norm for J_{id}, i.e.

$$\min_{\boldsymbol{\alpha}, \mathbf{u}_k^{\mathrm{f}}, \hat{\mathbf{y}}_k^{\mathrm{f}}} \quad J_{\mathrm{ctrl}}(\mathbf{u}_k^{\mathrm{f}}, \hat{\mathbf{y}}_k^{\mathrm{f}}) + \lambda \cdot \|\boldsymbol{\alpha}\|_1 \tag{19a}$$

$$\text{s.t.} \quad (8), \quad \hat{\mathbf{y}}_k^{\mathrm{f}} \in \mathcal{Y}, \quad \mathbf{u}_k^{\mathrm{f}} \in \mathcal{U}. \tag{19b}$$

Indeed, even for λ large, conditions (10) are not enforced, and the ℓ_1-norm regularization in (19) does not necessarily promote the selection of past trajectories that are close to the predicted one.

Fortunately, Proposition 1 also suggests a family of approaches to fulfil its assumptions. First, conditions (10) can be trivially embedded in (19) without making the problem more challenging to solve. Indeed, since (10) are convex constraints, they do not present any particular difficulty. The proximity condition resulting in a small $V_{\mathcal{D}}\left(\boldsymbol{\alpha}\right)$ is more demanding. A prototype problem to address that issue without regularization term can for example take the form

$$\min_{\boldsymbol{\alpha},\, \mathbf{u}_k^{\mathrm{f}},\, \hat{\mathbf{y}}_k^{\mathrm{f}}} \quad J_{\mathrm{ctrl}}(\mathbf{u}_k^{\mathrm{f}},\, \hat{\mathbf{y}}_k^{\mathrm{f}}) \tag{20a}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{u}_k^{\mathrm{p}} \\ \mathbf{y}_k^{\mathrm{p}} \\ \mathbf{u}_k^{\mathrm{f}} \\ \hat{\mathbf{y}}_k^{\mathrm{f}} \end{bmatrix} = \sum_{i \in \mathcal{D}} \alpha_i \begin{bmatrix} \mathbf{u}_i^{\mathrm{p}} \\ \mathbf{y}_i^{\mathrm{p}} \\ \mathbf{u}_i^{\mathrm{f}} \\ \mathbf{y}_i^{\mathrm{f}} \end{bmatrix} \tag{20b}$$

$$\sum_{i \in \mathcal{D}} \alpha_i = 1, \quad \alpha_i \geq 0 \tag{20c}$$

$$\max_{\mathbf{v}, \mathbf{w} \in C_{\mathcal{D}}(\boldsymbol{\alpha})} \|\mathbf{v} - \mathbf{w}\|^2 \leq \epsilon \tag{20d}$$

$$\hat{\mathbf{y}}_k^{\mathrm{f}} \in \mathcal{Y}, \quad \mathbf{u}_k^{\mathrm{f}} \in \mathcal{U} \tag{20e}$$

for a given ϵ, fixing a bound for $V_{\mathcal{D}}(\boldsymbol{\alpha})$. The difference between (20) and (19) comes through the constraints (20c) and (20d). The constraint (20c) comes from condition (10) and enforces the linear combination (4) to be a convex combination of past trajectories. Constraint (20d) is the proximity constraint that enforces the convex hull of the past trajectories that are selected to be close to the predicted one. This constraint turns out to be a challenging one, as it is combinatorial in nature, i.e. all combinations of past trajectories from which one can construct a convex combination satisfying (6a) such that the right-hand side of (20d) is less than ϵ must be found. Unfortunately, this is a computationally demanding problem, and we are not aware of any efficient algorithm to solve it. Building predictive control schemes based on (20) is therefore not feasible in practice.

In Sec. IV, we therefore propose a decomposition of (20) where a subset of relevant past trajectories satisfying (20d) is selected first, and then problem (20) is solved on that particular set, omitting constraint (20d). Before presenting that method, some further discussions on (20) are in order. First, the joint feasibility of constraints (20b) - (20d) can be a challenge. Indeed, for ϵ small, constraint (20d) can drastically restrict the set of admissible trajectories, denoted by a set of indices $\mathcal{S}$ in the following. Restricting $\mathcal{S}$ can make the linear combination (20b) difficult to satisfy in terms of matching $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$, especially when imposing (20c) to use convex combinations of the admissible trajectories. The feasibility of (20) then would in principle require a high density of past trajectories in $\mathcal{D}$ such that a basis of the past input-output sequences can be formed under constraint (20d). In practice, ensuring the feasibility of problem (20) would typically require a relaxation of (20b) - (20d). The relaxation of (20b) has already been investigated for the more classic problem (5), see [18]. It allows the solver to modify the initial conditions of the prediction to match the past input-output sequences. We will not discuss this relaxation here, as it does not pertain to our proposed method. Instead, we relax (20d) by first pre-selecting trajectories from $\mathcal{D}$ that meet the (20d) criteria. Then, we solve (20) using that refined set, omitting (20d).

A second useful observation is that for ϵ small, the output prediction formed in problem (20) is at a lower risk of being corrupted by the control objective J_{ctrl} than the classic formulation (5). Indeed, minimizing the control objective J_{ctrl} by corrupting the prediction requires (20) to exploit the error term (15) for reducing J_{ctrl}. However, as long as (20) remains feasible, that error term can be made arbitrarily small via ϵ, such that the distortion of the prediction for optimality can be directly managed.

Finally, it ought to be underlined here that the classic formulation (5) is a relaxation of (20) where (20c)-(20d) are omitted, and a regularization of $\boldsymbol{\alpha}$ used instead.

IV. Method

In this section, we introduce a computational method to tackle the combinatorial complexity of (20) arising from constraint (20d). Rather than attempting to solve the problem directly, we suggest constructing a subset $\mathcal{S}$ of past trajectories that adhere to constraint (20d). From there, we can formulate a relaxed version of the problem, without constraint (20d), which is amenable to standard optimization algorithms. Indeed, once a subset $\mathcal{S}$ of indices for relevant past trajectories is selected, the predictive control problem reads as:

$$\min_{\boldsymbol{\alpha},\, \hat{\mathbf{y}}_k^{\mathrm{f}},\, \mathbf{u}_k^{\mathrm{f}}} \quad J_{\mathrm{ctrl}}(\mathbf{u}_k^{\mathrm{f}},\, \hat{\mathbf{y}}_k^{\mathrm{f}}) \tag{21a}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{u}_k^{\mathrm{p}} \\ \mathbf{y}_k^{\mathrm{p}} \\ \mathbf{u}_k^{\mathrm{f}} \\ \hat{\mathbf{y}}_k^{\mathrm{f}} \end{bmatrix} = \sum_{i \in \mathcal{S}} \alpha_i \begin{bmatrix} \mathbf{u}_i^{\mathrm{p}} \\ \mathbf{y}_i^{\mathrm{p}} \\ \mathbf{u}_i^{\mathrm{f}} \\ \mathbf{y}_i^{\mathrm{f}} \end{bmatrix} \tag{21b}$$

$$\sum_{i \in \mathcal{S}} \alpha_i = 1, \qquad \alpha_i \geq 0 \tag{21c}$$

$$\hat{\mathbf{y}}_k^{\mathrm{f}} \in \mathcal{Y}, \quad \mathbf{u}_k^{\mathrm{f}} \in \mathcal{U}. \tag{21d}$$

Hence, constraint (20d) is eliminated by selecting a priori which past trajectories can be involved in the linear combination (21b). We also observe that the dimension of $\boldsymbol{\alpha}$ in (21) has the cardinality of $\mathcal{S} \subseteq \mathcal{D}$ which is typically much smaller than the one of $\boldsymbol{\alpha}$ in (20). In order to discuss the selection of $\mathcal{S}$, one needs to observe that (20d) has two requirements:

1) proximity to the recent past inputs and outputs $\mathbf{u}_k^{\mathrm{p}}, \mathbf{y}_k^{\mathrm{p}}$,
2) proximity to the future input sequence $\mathbf{u}_k^{\mathrm{f}}$.

Requirement 1 is fairly straightforward to satisfy, as it simply consists in selecting past trajectories $\mathbf{d}_i$ for which $\mathbf{u}_i^{\mathrm{p}}, \mathbf{y}_i^{\mathrm{p}}$ are proximal to the given $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$. This establishes a pool of past trajectories that are admissible in terms of $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$. The second requirement is more difficult as $\mathbf{u}_k^{\mathrm{f}}$ is defined by solving problem (20), and therefore not known a priori. In order to circumvent this difficulty, we propose to select among the admissible trajectories those that yield a low control cost J_{ctrl}. The expectation then is that trajectories conducive to building good control solutions will be selected, and that these trajectories will be close to each other, hence satisfying (20d). We develop these concepts more precisely in the following.

A. Proximal trajectories & proximity filter

The feature selection problem is to find a subset of the input-output data with sufficient proximity to the recent inputs and outputs $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$ over an initialization horizon T_{ini}. The metric to assess the proximity is dependent on the application, e.g. outputs evolving on Euclidean spaces can be compared using the Euclidean distance, while outputs evolving on manifolds can be more complex. In this paper, we measure proximity as the squared Euclidean distance between the recent inputs and outputs $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$ and the corresponding past inputs and outputs $\mathbf{u}_i^{\mathrm{p}}$, $\mathbf{y}_i^{\mathrm{p}}$, and build the set of proximal trajectories $\mathcal{P} \subseteq \mathcal{D}$ as:

$$\mathcal{P} = \left\{ i \in \mathcal{D} \quad \text{s.t.} \quad \left\| \begin{bmatrix} \mathbf{y}_i^{\mathrm{p}} \\ \mathbf{u}_i^{\mathrm{p}} \end{bmatrix} - \begin{bmatrix} \mathbf{y}_k^{\mathrm{p}} \\ \mathbf{u}_k^{\mathrm{p}} \end{bmatrix} \right\|^2 \leq \epsilon^{\mathrm{p}} \right\} \tag{22}$$

for some $\epsilon^{\mathrm{p}} > 0$. For applications with fast dynamics that require high sampling rates (e.g. robotics), and with a significant amount of available data, the set $\mathcal{D}$ can be very large, which can be a challenge for the online computation of $\mathcal{P}$. However, as this is a computer science problem, we resort to efficient algorithms to compute the proximal set $\mathcal{P}$ in a reasonable amount of time.

B. High-performance trajectories & performance filter

Among the proximal trajectories $\mathcal{P}$, we need to select those that are meaningful to serve as a basis for the predictive control problem. To that end, we propose to base the selection on the performance of the past trajectories in terms of the control objective J_{ctrl}. More specifically, we propose to order the proximal past trajectories in terms of $J_{\mathrm{ctrl}}\left(\mathbf{y}_i^{\mathrm{f}}, \mathbf{u}_i^{\mathrm{f}}\right)$. Let us label the index of the trajectory with the lowest control cost as $i^\star$ given by:

$$i^\star = \arg\min_{i \in \mathcal{P}} J_{\mathrm{ctrl}}\left(\mathbf{y}_i^{\mathrm{f}}, \mathbf{u}_i^{\mathrm{f}}\right). \tag{23}$$

We can then construct the set of indices $\mathcal{S} \subseteq \mathcal{P} \subseteq \mathcal{D}$ as

$$\mathcal{S} = \left\{ i \in \mathcal{P} \quad \text{s.t.} \quad \|\mathbf{u}_i^{\mathrm{f}} - \mathbf{u}_{i^\star}^{\mathrm{f}}\|^2 \leq \epsilon^{\mathrm{f}} \right\} \tag{24}$$

for some $\epsilon^{\mathrm{f}} > 0$. Given that $\mathbf{y}_i^{\mathrm{f}}$ are constrained by $\mathbf{y}_i^{\mathrm{f}} - \phi(\mathbf{u}_i^{\mathrm{f}}, \mathbf{u}_i^{\mathrm{p}}, \mathbf{y}_i^{\mathrm{p}}) = 0$, they are not used for the construction of $\mathcal{S}$. The set $\mathcal{S}$ contains the indices of high-performance trajectories that will be used to build the predictive control solution. Given Proposition 1, we can now state the following result that establishes a bound on the suboptimality of the predictive control solution with respect to the control objective J_{ctrl} that is due to the approximation error $\mathbf{e}_\phi(\mathcal{S}, \alpha)$.

Proposition 2: Assume:

1) $\epsilon^{\mathrm{f}} + \epsilon^{\mathrm{p}} \leq \epsilon$
2) The mixed objective J is locally Lipschitz continuous on $\mathcal{S}$ with respect to variations in $\mathbf{y}$, i.e. $\left\|J\left(\mathbf{y}_i^{\mathrm{f}}, \cdot\right) - J\left(\mathbf{y}_j^{\mathrm{f}}, \cdot\right)\right\| \leq c \left\|\mathbf{y}_i^{\mathrm{f}} - \mathbf{y}_j^{\mathrm{f}}\right\| \forall i, j \in \mathcal{S}$.
3) Let us denote

$$\Lambda_k = \left\{ \boldsymbol{\alpha} \quad \text{s.t.} \quad \begin{bmatrix} \mathbf{u}_k^{\mathrm{p}} \\ \mathbf{y}_k^{\mathrm{p}} \end{bmatrix} = \sum_{i \in \mathcal{S}} \alpha_i \begin{bmatrix} \mathbf{u}_i^{\mathrm{p}} \\ \mathbf{y}_i^{\mathrm{p}} \end{bmatrix} \quad \text{and} \quad (21c) \right\} \tag{25}$$

and assume that the optimal control input $\mathbf{u}_k^\star$ at time k is such that

$$\mathbf{u}_k^\star \in \left\{ \sum_{i \in \mathcal{S}} \alpha_i \mathbf{u}_i^{\mathrm{f}} \quad \text{s.t} \quad \boldsymbol{\alpha} \in \Lambda_k \right\}. \tag{26}$$

Let $\boldsymbol{\alpha}^\star$, $\hat{\boldsymbol{\alpha}}^\star \in \Lambda_k$ be the minimizers of the respective cost functions:

$$\min_{\boldsymbol{\alpha} \in \Lambda_k} \mathcal{J}_{\mathcal{S}}(\boldsymbol{\alpha}) := \min_{\boldsymbol{\alpha} \in \Lambda_k} J\left(\phi\left(\sum_{i \in \mathcal{S}} \alpha_i \mathbf{d}_i\right), \sum_{i \in \mathcal{S}} \alpha_i \mathbf{u}_i\right), \tag{27}$$

$$\min_{\boldsymbol{\alpha} \in \Lambda_k} \hat{\mathcal{J}}_{\mathcal{S}}(\boldsymbol{\alpha}) := \min_{\boldsymbol{\alpha} \in \Lambda_k} J\left(\sum_{i \in \mathcal{S}} \alpha_i \phi(\mathbf{d}_i), \sum_{i \in \mathcal{S}} \alpha_i \mathbf{u}_i\right). \tag{28}$$

Then the following inequality holds:

$$\mathcal{J}_{\mathcal{S}}(\hat{\boldsymbol{\alpha}}^\star) \leq \mathcal{J}_{\mathcal{S}}(\boldsymbol{\alpha}^\star) + 2c\epsilon \cdot \max_{\mathbf{v} \in C_{\mathcal{S}}(\boldsymbol{\alpha})} \left\|\nabla^2 \phi(\mathbf{v})\right\|. \tag{29}$$

Proof: Assumption 3 implies that the optimal control input $\mathbf{u}_k^\star$ at time k is such that

$$\min_{\boldsymbol{\alpha} \in \Lambda_k} \mathcal{J}_{\mathcal{S}}(\boldsymbol{\alpha}) = \min_{\mathbf{u}_k^{\mathrm{f}}} J\left(\varphi(\mathbf{u}_k^{\mathrm{f}}, \mathbf{u}_k^{\mathrm{p}}, \mathbf{y}_k^{\mathrm{p}}), \mathbf{u}_k^{\mathrm{f}}\right) \tag{30}$$

holds. The corresponding minimizer satisfies $\boldsymbol{\alpha}^\star \in \Lambda_k$ by construction of Assumption 3. Similar arguments hold for $\hat{\boldsymbol{\alpha}}^\star \in \Lambda_k$. Using Proposition 1 and the continuity Assumption 2, we observe that

$$\begin{aligned} \left|\hat{\mathcal{J}}_{\mathcal{S}}(\boldsymbol{\alpha}) - \mathcal{J}_{\mathcal{S}}(\boldsymbol{\alpha})\right| &\leq c \left\|e_\phi(\mathcal{S}, \boldsymbol{\alpha})\right\| \\ &\leq c \cdot \max_{\mathbf{v} \in C_{\mathcal{S}}(\boldsymbol{\alpha})} \left\|\nabla^2 \phi(\mathbf{v})\right\| V_{\mathcal{S}}(\boldsymbol{\alpha}) \\ &\leq c\epsilon \cdot \max_{\mathbf{v} \in C_{\mathcal{S}}(\boldsymbol{\alpha})} \left\|\nabla^2 \phi(\mathbf{v})\right\| \end{aligned} \tag{31}$$

holds for all $\boldsymbol{\alpha} \in \Lambda_k$. The final result then follows as

$$\begin{aligned} \mathcal{J}_{\mathcal{S}}(\hat{\boldsymbol{\alpha}}^\star) &\leq \hat{\mathcal{J}}_{\mathcal{S}}(\hat{\boldsymbol{\alpha}}^\star) + c\epsilon \cdot \max_{\mathbf{v} \in C_{\mathcal{S}}(\boldsymbol{\alpha})} \left\|\nabla^2 \phi(\mathbf{v})\right\| \\ &\leq \hat{\mathcal{J}}_{\mathcal{S}}(\boldsymbol{\alpha}^\star) + c\epsilon \cdot \max_{\mathbf{v} \in C_{\mathcal{S}}(\boldsymbol{\alpha})} \left\|\nabla^2 \phi(\mathbf{v})\right\| \\ &\leq \mathcal{J}_{\mathcal{S}}(\boldsymbol{\alpha}^\star) + 2c\epsilon \cdot \max_{\mathbf{v} \in C_{\mathcal{S}}(\boldsymbol{\alpha})} \left\|\nabla^2 \phi(\mathbf{v})\right\|. \end{aligned} \tag{32}$$

■

Remark 1: Assumption 3 in Proposition 2 imposes the requirement that the initial data $\mathbf{u}_k^{\mathrm{p}}$, $\mathbf{y}_k^{\mathrm{p}}$ can be reproduced by linear combinations of $\mathbf{u}_i^{\mathrm{p}}$, $\mathbf{y}_i^{\mathrm{p}}$ with $i \in \mathcal{S}$. This is a natural requirement in the context of the proposed approach, but it entails that the span of the stored trajectories has to exactly match the trajectory space of the system. Noise in the data or a small set of trajectories $\mathcal{S}$ may thus lead to Λ_k being empty. The problem is commonly tackled by relaxing the second block row in (21b), see e.g. [7].

V. Numerical Example

We test our method on the inverted cart-pendulum system. It is described by the state vector $\mathbf{x}$, output vector $\mathbf{y}$, and input $u = F$, where $\mathbf{x} = [p, \theta, v, \omega]^\top$ represents the position of the cart p, the angle of the pendulum θ, and their respective velocities $\dot{x} = v$ and $\dot{\theta} = \omega$. We consider the positions as output, i.e. $\mathbf{y} = [p, \theta]^\top$. The accelerations of the inverse pendulum are governed by:

$$\dot{v} = \frac{-ml\sin(\theta)\dot{\theta}^2 + mg\cos(\theta)\sin(\theta) + u}{M + m(1-\cos(\theta)^2)}$$

$$\dot{\omega} = \frac{-ml\cos\theta\sin\theta\dot{\theta}^2 + F\cos(\theta) + (M+m)g\sin(\theta)}{l(M + m(1-\cos(\theta)^2))} .$$

The control objective is to stabilize the system around the unstable upward equilibrium point with the cart position at the origin, which corresponds to $\mathbf{x}_{eq} = \mathbf{0}_{4\times 1}$. The force input is constrained to $|F| \leq 15\,\mathrm{N}$.

We use [19] to generate a library of 2000 swing-up trajectories to build the predictor. The initial conditions for the trajectories are sampled from uniform distributions with $x_0 \sim \mathcal{U}(-0.2, 0.2)\,\mathrm{m}$ and $\theta_0 \sim \mathcal{U}(150, 210)\,\mathrm{deg}$, with initial velocities and control input set to zero. The library of trajectories is shown in Fig. 2. Including additional trajectories with poor swing-up performance in the library does not change the results, as they are not selected by the filter.

To solve the OCP for the data-driven approach, we use IPOPT [20]. The weighting matrices for the control objective are $\mathbf{Q} = \mathrm{diag}(100, 100)$ and $\mathbf{R} = 0.1$, with initialization horizon and prediction horizon set to $T_{\mathrm{ini}} = 10$ and $N = 50$, respectively. The proximity filter is applied to each output/input channel separately, which allows using different values for ϵ^{p}. In the simulation we use $\epsilon^{\mathrm{p}}_x = 0.05\,\mathrm{m}$ for the cart position, $\epsilon^{\mathrm{p}}_\theta = 2\,\mathrm{deg}$ for the pendulum angle, and $\epsilon^{\mathrm{p}}_u = 5\,\mathrm{N}$ for the control input.

We picked one trajectory from the library and use the rest of the library for predictions. The dimension of $\boldsymbol{\alpha}$ is set to 20, i.e. after the proximity filter step on the initial conditions, the library is reduced to 20 trajectories that include the optimal trajectory and its 19 nearest neighbors as shown in Fig. 2. The resulting predictions are shown in Fig. 3 and compared to the rollouts of the optimal control sequence which only start to diverge at $t = 50$ for θ, i.e. after 40 prediction steps. This shows that the predictions are not being corrupted by the majority of the library trajectories that represent a faster swing-up.

We implemented a DeePC-like approach using the same library. However, we did not use the prefiltered + ranking approach and instead used all available trajectories directly. We attempted to use ℓ_1-norm regularization, but we were unable to find a suitable value for the regularization weight λ. As a result, we were unable to obtain predictions that were usable for control, and therefore, we will not report them here.

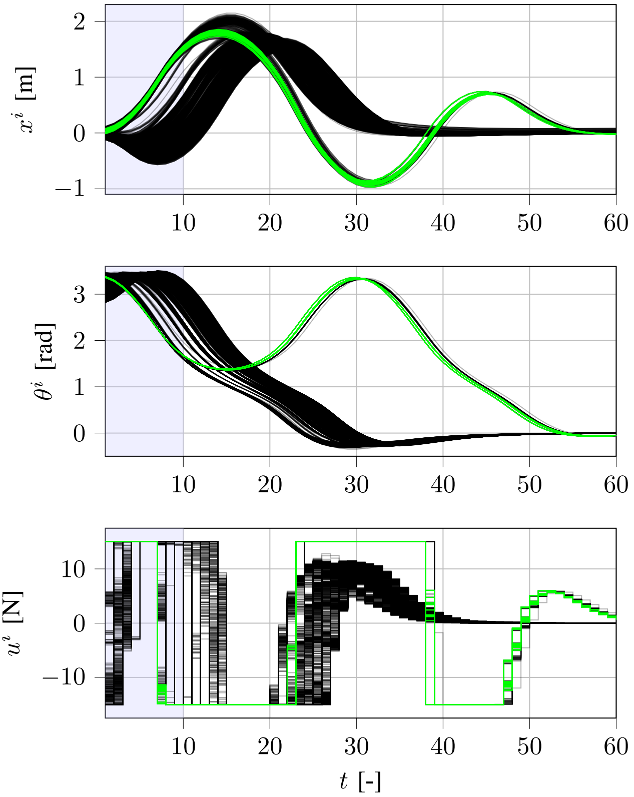

Fig. 2. The green trajectories were selected from all possible trajectories to build the predictor, while the remaining trajectories are shown in black. It is important to note that the dataset of all libraries is much larger and distinct from the trajectories used to build the predictor. The blue rectangle in the figure corresponds to the initial conditions, which consist of the first $T_{\mathrm{ini}} = 10$ samples.

VI. Conclusion

In this paper, we proposed an extension of recent data-driven predictive control approaches to handle nonlinear systems. Our focus was on understanding the impact of nonlinearities on the predictive control formulation, and we proposed a feature selection algorithm to choose a subset of available trajectories that represent the system behavior with minimal prediction error. To validate the effectiveness of our approach, we applied the algorithm to a nonlinear system and demonstrated its ability to produce accurate predictions. Overall, our work provides a promising solution for addressing nonlinearities in data-driven predictive control.

References

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

[2] W. Favoreel, B. D. Moor, and M. Gevers, "SPC: Subspace Predictive Control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004–4009, Jul. 1999.

[3] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, Dec. 2008.

[4] H. Yang and S. Li, "A data-driven predictive controller design based on reduced hankel matrix," in *2015 10th Asian Control Conference (ASCC)*, 2015, pp. 1–7.

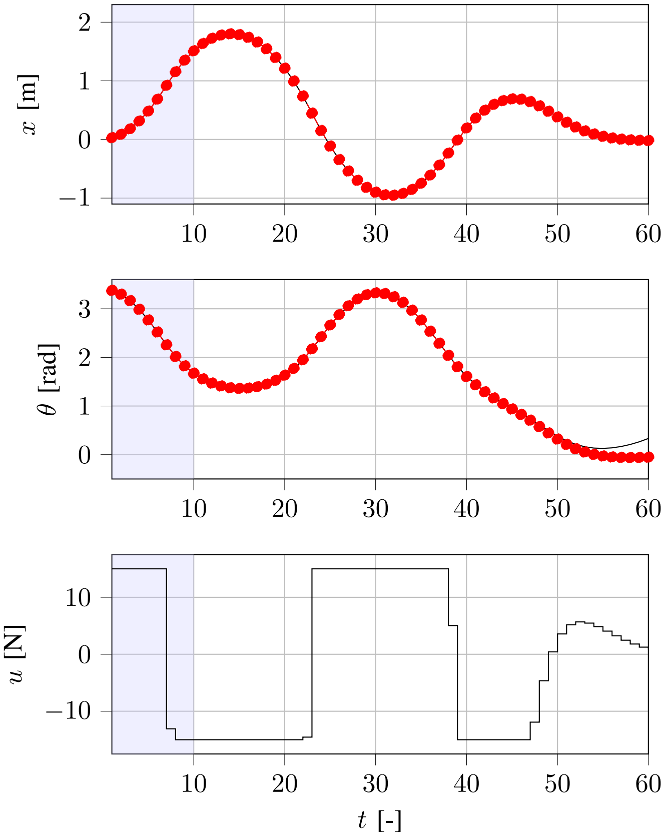

Fig. 3. The figure illustrates our approach of using filtered and ranked trajectories to create a predictor. The first two plots display the red dotted predictions overlaid on the solid black rollouts for the optimal control sequence in the third plot. The blue rectangle in the figure corresponds to the initial conditions, which consist of the first $T_{\text{ini}} = 10$ samples.

[5] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, Apr. 2005.

[6] J. Coulson, J. Lygeros, and F. Dörfler, "Data-Enabled Predictive Control: In the Shallows of the DeePC," in *2019 18th European Control Conference (ECC)*, Jun. 2019, pp. 307–312.

[7] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2021.

[8] ——, "Linear tracking mpc for nonlinear systems - part ii: The data-driven case," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4406–4421, Sep. 2022.

[9] J.-W. van Wingerden, S. P. Mulders, R. Dinkla, T. Oomen, and M. Verhaegen, "Data-enabled predictive control with instrumental variables: the direct equivalence with subspace predictive control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, Dec 2022, pp. 2111–2116.

[10] F. Fiedler and S. Lucia, "On the relationship between data-enabled predictive control and subspace predictive control," in *2021 European Control Conference (ECC)*, June 2021, pp. 222–229.

[11] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct & indirect data-driven control formulations via regularizations and relaxations," *IEEE Transactions on Automatic Control*, pp. 1–1, 2022.

[12] P. Mattsson and T. B. Schön, "On the regularization in DeePC," in *IFAC World Congress*, 2023.

[13] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory - Implementation - Applications*. Springer Science & Business Media, 2012.

[14] R. Dinkla, S. P. Mulders, J. W. van Wingerden, and T. A. Oomen, "Closed-loop aspects of data-enabled predictive control," in *IFAC 22st Triennial World Congress*, 2023.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[16] L. D. Natale, Y. Lian, E. T. Maddalena, J. Shi, and C. N. Jones, "Lessons learned from data-driven building control experiments: Contrasting gaussian process-based mpc, bilevel deepc, and deep reinforcement learning," *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 1111–1117, 2022.

[17] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[18] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Robust data-enabled predictive control: Tractable formulations and performance guarantees," 2021.

[19] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados – a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, Mar. 2022.

[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006.