

# Choose Wisely: Data-driven Predictive Control for Nonlinear Systems Using Online Data Selection

**Joshua Näf**  
**Keith Moffat**  
**Jaap Eising**  
**Florian Dörfler**

*ETH Zurich, Zurich, Switzerland*

NAEFJO@ETHZ.CH  
 KMOFFAT@ETHZ.CH  
 JEISING@ETHZ.CH  
 DORFLER@ETHZ.CH

**Editors:** G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

## Abstract

This paper proposes Select-Data-driven Predictive Control (Select-DPC), a new method for controlling nonlinear systems using output-feedback for which data are available but an explicit model is not. At each timestep, Select-DPC employs only the most relevant data to implicitly linearize the dynamics in “trajectory space.” Then, taking user-defined output constraints into account, it makes control decisions using a convex optimization. This data-driven optimal control is applied in a receding-horizon manner. As the online data-selection is the core of Select-DPC, we propose and compare norm-based and manifold-embedding-based data selection methods. We evaluate Select-DPC on three benchmark nonlinear system simulators—rocket-landing, a robotic arm, and cart-pole inverted pendulum swing-up—comparing them with standard Data-enabled Predictive Control (DeePC) and Time-Windowed DeePC methods, and find that Select-DPC outperforms both methods. The source code can be found at: <https://github.com/naefjo/choose-wisely-paper>

**Keywords:** Data-driven Predictive Control, Data-enabled Predictive Control, Nonlinear Systems, Nonlinear Predictive Control, Manifold Embedding, Sequential Quadratic Programming

## 1. Introduction

The data-driven control and Reinforcement Learning (RL) communities are focused on controlling systems for which data is available, but a system model is not. Specifically, the control community has developed interest in “direct” Data-driven Predictive Control (DPC) methods [Dörfler et al. \(2023\)](#) as they reduce system modeling/identification time and effort and bring the benefits of predictive control to the “model-free” setting.

This paper proposes Select-DPC, a direct DPC method for nonlinear systems. Select-DPC extends DPC to the nonlinear setting by iteratively optimizing a convex Quadratic Program (QP) over an implicit linearization in “trajectory space” constructed from data. We define trajectory space as the high-dimensional space in which each dimension corresponds to an input or output over the course of a discrete-time, finite-length trajectory. Linearizing in trajectory space is achieved by selecting only trajectories that are “close” to the operating point in trajectory space.

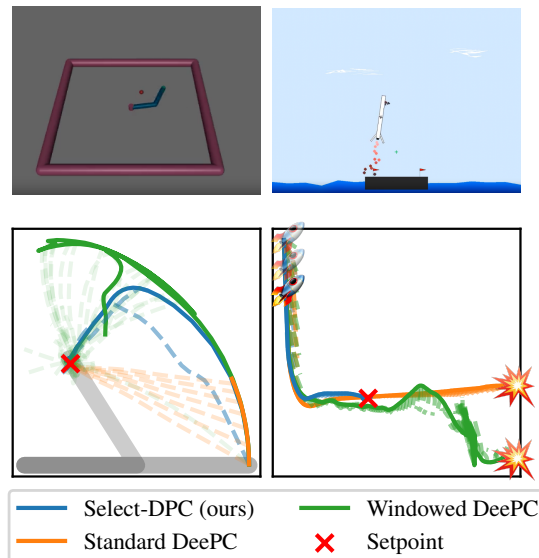


Figure 1: Closed-loop trajectories of Select-DPC (ours), standard DeePC, and Windowed DeePC in the Reacher (*left*) and the Rocket Lander (*right*) environments, as well as the corresponding open-loop predictions (dashed). Only Select-DPC converges to the setpoint.

**Literature Review** Output-constrained systems are challenging for RL methods, which learn the constraints by exhaustive trial and error [Sutton and Barto \(1998\)](#). Such systems have traditionally been the domain of Model Predictive Control (MPC) [Morari and Lee \(1999\)](#), which solves a receding-horizon optimal control problem based on a state-space representation with an online optimization. In the nonlinear domain, the optimal control problem can be solved using Sequential Quadratic Programming (SQP). SQP-MPC iteratively linearizes the dynamics around a solution estimate, solves a QP around the linearization, and updates the solution estimate [Diehl et al. \(2002\)](#).

The class of Data-driven Predictive Controllers construct feasible trajectories of a linear, time-invariant (LTI) system purely from data. These methods bypass the sequential *indirect* system identification and MPC design pipeline and are hence called *direct methods*. Data-enabled Predictive Control (DeePC) [Coulson et al. \(2019\)](#), for instance, linearly combines input-output trajectories in the data set. Such predictions fit into the behavioral control framework [Willems \(1991\)](#). For a survey on DeePC we refer the reader to [Markovsky et al. \(2023\)](#). DPC methods have demonstrated impressive results on various control tasks. While DPC methods can handle minor violations of the LTI assumption by treating them as additive measurement noise, the LTI restriction is severe and leads to arbitrary performance loss on generic nonlinear systems.

Several approaches to DeePC in the nonlinear case can be found in the literature. [Berberich and Allgöwer \(2024\)](#) provide tailored formulations of the Fundamental Lemma [Willems et al. \(2005\)](#) to specific classes of nonlinear systems. These formulations enable efficient synthesis of DPC for systems belonging to these specific system classes. However, this already imposes a strong prior on the global nonlinear structure of the dynamical process, which might not be available for an arbitrary system. Alternatively, [Berberich et al. \(2022\)](#) propose using a set of the most recent past observations in a sliding window fashion along the current closed-loop trajectory to form the basis of a linear predictor, herein referred to as Time-Windowed DeePC. As this method only uses recent closed-loop data, it is practical only for systems with benign nonlinearities and with high signal-to-noise ratios. Approaches in the literature also suggest to reformulate the predictor in a kernelized fashion, enabling the use of nonlinear kernels [Lian and Jones \(2021\)](#); [Huang et al. \(2024\)](#), using nonlinear basis functions or reformulating the problem in a lifted state-space using Koopman operators [Lazar \(2023\)](#). While these methods offer nonlinear function approximation, the resulting optimization becomes high-dimensional, nonlinear, and generally nonconvex, requiring increased online computational complexity and resulting in suboptimal outcomes.

In contrast with the optimization-based MPC literature, sampling-based MPC methods generate an ensemble of roll-outs by randomly sampling control inputs [Nagabandi et al. \(2018\)](#) and then applying an iterative refinement to obtain near optimal control sequences [Botev et al. \(2013\)](#); [Williams et al. \(2017\)](#). These methods are favored in domains such as RL, as sampling-based MPC only requires zero-order model information, i.e. no gradient of the model, making them computationally cheap to evaluate. However, sampling-based MPC usually has no structured way of incorporating output constraints, since they rely on model roll-outs, and require the user to offload this complexity

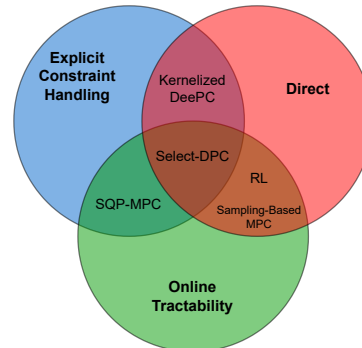


Figure 2: Venn diagram illustrating how the discussed methods fall within the *Explicit Constraint Handling*, *Direct*, and *Online Tractability* merit categories. Select-DPC has all three merits.

into other parts of the method, such as the model itself or the cost. In the context of RL, optimization-based planning techniques with explicit, learning-based models suffer from accumulating prediction errors [Moerland et al. \(2022\)](#), exploding gradients due to backpropagation through time or model exploitation in regions of low data density [Kurutach et al. \(2018\)](#). Figure 2 shows a conceptual overview of the mentioned strategies and their respective traits.

The Select-DPC data-selection methods rely on finding high-dimensional nearest-neighbors, a concept that has been well established and applied in adjacent fields such as SLAM for feature matching and pose optimization [Bailey and Durrant-Whyte \(2006\)](#), 3D shape registration [Besl and McKay \(1992\)](#) and in the context of dimensionality reduction and manifold learning [Jia et al. \(2022\)](#).

**Contributions** The contributions of this paper are threefold:

1. We introduce *Select-DPC*, a data-driven quadratic program predictive controller for nonlinear systems which implicitly linearizes the system dynamics in trajectory space by selecting and using only the most relevant data at each decision moment.
2. We propose two data-selection methods to achieve linearization in trajectory space, a norm-based method and a manifold-embedding-based method.
3. We demonstrate the benefits of Select-DPC on three benchmark nonlinear systems including a planar rocket-landing, a two degree of freedom robotic arm, and a cart-pole inverted pendulum, comparing it to standard DeePC and a Time-Windowed DeePC approach and find that Select-DPC outperforms both in terms of closed-loop cost and prediction accuracy.

## 2. Background

**Preliminaries** We concern ourselves with nonlinear systems, for which we assume an autoregressive form  $y_f(k) = f(y_p(k), u_p(k), u_f(k))$  exists. We use  $y \in \mathbb{R}^p$  and  $u \in \mathbb{R}^m$  to denote measurements and inputs of the nonlinear system respectively. The subscripts  $a_p$  and  $a_f$  denote past and future values of a variable  $a$  and are defined as  $a_p(k) := \{a(k-i)\}_{i=T_p-1}^0$  and  $a_f(k) := \{a(k+i)\}_{i=1}^{T_f}$ .

**Data-Enabled Predictive Control (DeePC)** Select-DPC builds upon linear DPC which solves a receding-horizon predictive control problem subject to input and output constraints. The novelty of DPC is that it achieves predictive control purely based on implicit predictions of input-output data without ever constructing an explicit predictor (a model) of future measurements given a set of inputs. DeePC [Coulson et al. \(2019\)](#), for instance, solves the following optimization problem

$$\min_{y,u,g} \sum_{i=1}^{T_f} c_i(y_i, u_i) + r(g), \text{ s.t. } Hg = \begin{bmatrix} u_p^\top, y_p^\top, u^\top, y^\top \end{bmatrix}^\top, (u_i, y_i) \in \mathcal{U} \times \mathcal{Y} \forall i = \{1, \dots, T_f\}, \quad (1)$$

where each column in the matrix  $H$  represents a  $L$ -length trajectory from a pre-collected data set  $\mathcal{D} = \{\tau_0, \dots, \tau_{n_d}\}$ , with  $\tau = \{(u_i, y_i)\}_{i=1}^L$ .  $H$  can be partitioned into  $[H_p^\top, H_f^\top]^\top$ , where each factor represents the partition of the data set into past and future quantities respectively. Several regularizer  $r(g)$  have been introduced to further bias the choice of said linear combinations [Dörfler et al. \(2023\)](#). We refer to [Markovsky et al. \(2023\)](#) or Appendix A for a detailed discussion of DeePC.

Solving DeePC online using a large data set of trajectories is intractable as the solve time of the optimization problem scales superlinearly with  $\text{card}(\mathcal{D})$ . Furthermore, in the presence of nonlinearities, the LTI assumption of DeePC is violated, resulting in suboptimal performance.

### 3. Select-DPC

We now introduce *Select-DPC*, an algorithm that allows us to solve a nonlinear data-driven receding-horizon optimal control problem using a convex quadratic programming routine by choosing, at each solver iteration, a subset of trajectories  $\tilde{\mathcal{D}}$  with cardinality  $\text{card}(\tilde{\mathcal{D}}) = N_{\text{cols}}$  from  $\mathcal{D}$  where  $N_{\text{cols}} \ll \text{card}(\mathcal{D})$ . Algorithm 1 summarizes the implementation of Select-DPC. In between two real-time sampling instances, an iterative procedure is performed where, at every iteration, we

select from a data set  $\mathcal{D}$  a subset of trajectories  $\tilde{\mathcal{D}}$  that are “relevant” to the current linearization point given by the open-loop solution  $\tilde{\tau}$ . While we loosely use the term relevance here, this mechanism will be discussed in more detail in Section 4. Select-DPC then solves a DPC problem using the subset of data. This is iterated until some convergence criterion, e.g. specified

---

#### Algorithm 1 Select-DPC

---

```

1: function Select-DPC( $u_p, y_p$ )
2:   while convergence criterion not met do
3:      $\tilde{\tau} \leftarrow \text{DPC.getLastPrediction}()$ 
4:      $\tilde{\mathcal{D}} \leftarrow \text{select}(\mathcal{D}, \tilde{\tau}, N_{\text{cols}})$ 
5:      $u \leftarrow \text{DPC.genAction}(u_p, y_p, \tilde{\mathcal{D}})$ 
6:   end while
7:   Return  $u_0$ 
8: end function

```

---

tolerance or maximum number of iterations, is reached. At the first time step, the open-loop trajectory is warm-started by uniformly replicating the initial system measurement across the horizon, with zero inputs assumed. Notice that there is no restriction on the choice of DPC solver. We now discuss a number of important attributes of this proposed method.

**Cost-agnostic** Since the data set generation is structurally independent of the control formulation, a posteriori cost or constraint modification is possible, allowing the method to be applied zero-shot to new problems without the need for additional data collection.

**Hyperparameters** The performance of Select-DPC depends on the tuning of several hyperparameters. First, the DPC specific hyperparameters, such as regularization terms and horizon lengths  $T_p, T_f$ . These can either be hand-tuned according to established heuristics Elokda et al. (2021) or optimized through gradient-descent methods Cummins et al. (2024) or black-box parameter tuning Berkenkamp et al. (2016). For DPC tuning guidelines, we refer to Markovsky et al. (2023). Select-DPC adds onto these the design parameters  $N_{\text{cols}}$  as well as the method of data selection. In the following section, we will investigate two different data-selection methods which subsample a specified number of trajectories from a large offline library of trajectories observed during the data gathering process.

**Sequential Linearization** The iterative refinement of the solution until convergence is comparable to the model-based SQP-MPC algorithm for nonlinear systems Diehl et al. (2002). We note that Select-DPC’s implicit linearization is performed in trajectory space. In the model-based case, linearizing in trajectory space corresponds to a linear time-varying parametrization of the nonlinear dynamics for the MPC. This linear, time-varying parametrization contrasts with an LTI formulation in which the Jacobians are only evaluated at the current measurement  $y(k)$  and kept fixed along the horizon. See Appendix B for more details.

### 4. Data Selection

At the heart of Select-DPC lies the data selection method, which selects data points (representing trajectories  $\tau_i$ ) from a large (offline) data set. Since the subproblem solved by Select-DPC relies on linearly combining trajectories, the subset of data should exhibit predominantly linear dynamics. The suggested approach in the literature is to use a sliding window over past input-output measurements. We propose that  $\tilde{\mathcal{D}}$  should be chosen according to some *spatial distance metric* (in trajectory space) as opposed to *temporal proximity* (such as a sliding window) to the current linearization point  $\tilde{\tau}$ .

In the following, we will present norm-based and manifold-embedding-based selection, while placing emphasis on the modular nature of Select-DPC. While norm-based selection is efficient to implement and allows the user to easily update the data set online, manifold-embedding-based selection enables the distance computation in a lower dimensional embedding space which lessens the impact of the *curse of dimensionality* but comes with increased offline computation cost.

**Norm-based selection** Norm-based data selection selects from a data set  $\mathcal{D}$  the  $N_{\text{cols}}$  closest (in space) trajectories by computing the norm of the differences between the open-loop solution  $\tilde{\tau}$  and the trajectories in the data set  $\mathcal{D}$ . An implementation is given in Algorithm 2.

---

**Algorithm 2** Norm-based Data Selection
 

---

- 1: **function** NormDataSelection( $\mathcal{D}, \tilde{\tau}, N_{\text{cols}}$ )
  - 2:    $\tilde{\mathcal{D}} = \text{sort}(\{\|\tau_i - \tilde{\tau}\| \mid i = 1, \dots, n_d\})$
  - 3:   Pick first  $[1, N_{\text{cols}}]$  from  $\tilde{\mathcal{D}}$
  - 4: **end function**
- 

While this approach is intuitive, it is common knowledge that norm-based distances fall flat in high dimensional spaces due to the *curse of dimensionality* Aggarwal et al. (2001). Norm-based selection computes distances of trajectories, i.e. vectors with dimension  $(T_p + T_f)(m + p)$  and is hence also susceptible to this curse.

**Low Dimensional Representation using Manifold Learning**

The field of manifold learning (nonlinear dimensionality reduction) concerns itself with the compression of high-dimensional data onto manifolds with lower intrinsic dimension than the original data by using (nonlinear) projection techniques Lee and Verleysen (2007); Jia et al. (2022). Manifold Learning techniques rely on the *manifold hypothesis* Fefferman et al. (2016), which postulates that the intrinsic dimensionality of a data set  $\mathcal{D}$  is much lower than the dimensionality of the individual data points. For instance, if the system under consideration is a controllable and deterministic LTI system, then Willems' Fundamental Lemma states that the trajectories span a manifold (in particular a subspace) that can at most have  $n + T_f \cdot m$  degrees of freedom Willems et al. (2005), which is generally less than the trajectory dimensionality  $(T_p + T_f)(m + p)$ . Here,  $n$  denotes the dimension of the process's latent state,  $p$  is the dimension of the measurement vector and  $m$  indicates the dimension of the input signal. Motivated by this observation, we wish to find a lower dimensional representation of the data where the similarities can be computed without incurring the curse of dimensionality.

---

**Algorithm 3** Manifold-Embedding Data Selection
 

---

- 1: **function** ManifoldDataSel( $\mathcal{D}, \tilde{\tau}, N_{\text{cols}}$ )
  - 2:    $\mathcal{D}_e \leftarrow \text{embed}(\mathcal{D})$
  - 3:    $\tilde{\tau}_e \leftarrow \text{embed}(\tilde{\tau})$
  - 4:    $\tilde{\mathcal{D}}_e = \text{sort}(\{\|\tau_{e,i} - \tilde{\tau}_e\| \mid i = 1, \dots, n_d\})$
  - 5:   Pick first  $[1, N_{\text{cols}}]$  from  $\tilde{\mathcal{D}}_e$
  - 6: **end function**
- 

Isomap, introduced by Tenenbaum et al. (2000), is an unsupervised manifold learning method which aims to find an embedding of the data in a Euclidean space which preserves geodesic distances of a neighborhood graph of the data constructed in the data space. This comes with increased computational cost compared to norm based data selection but can offer inter data point distances which more accurately reflect the nonlinear nature of the data manifold in trajectory space. Also notice that the bulk of this added cost is a one-time upfront cost (computing the embedding), which can be performed offline.

Although this does not fully resolve the problem caused by the curse of dimensionality as the embedding dimension, i.e. the dimension where we need to compute distances, will still scale with at least with  $n + T_f \cdot m$ , it does offer a structured way of compressing the data to a reasonable dimension. Experiments showed that embedding dimensions lower than the minimum dimension required to represent a linear system already lead to diminishing returns in reconstruction error.

The pseudocode for manifold-embedding-based data selection using the Isomap algorithm is outlined in Algorithm 3. For algorithmic details of Isomap (i.e. the `embed` step) we refer to Tenenbaum et al. (2000) or Appendix C. Notice that the embedding of the data set in line 2 can be cached/computed offline. Online, the algorithm then proceeds to embed the query trajectory  $\tilde{\tau}_i$  and computes its similarities *in the lower dimensional embedding space*. Then, similarly to Algorithm 2, the relevant trajectories are returned in the trajectory space, after which the implicit predictor for DPC is constructed using the retrieved non-embedded data points.

## 5. Results and Discussion

Simulation experiments were performed to validate the efficacy of Select-DPC and to benchmark it against standard DeePC Coulson et al. (2019), which utilizes the entire data set, as well as the Time-Windowed DeePC variant proposed by Berberich et al. (2022), which constructs the implicit predictor by continuously updating the data set with new input-output data and discarding the most out-of-date measurement. All experiments were performed on an Apple MacBook Pro with an M2 Pro chip and 16GB of RAM. We use a maximum number of iterations  $n_{\max}$  as a stopping criterion. No process or measurement noise is added to the simulated systems. Hyperparameters are provided in Appendix D. In the experiments focusing on closed-loop cost (Figure 4 left and Figure 5 right), we performed an LQ decomposition of the data matrix Breschi et al. (2023) for data subsets larger than 400. While this drastically speeds up computation times, we did not observe significant differences in control performance compared to no LQ decomposition.

### 5.1. Landing a Reusable Rocket

We use a planar vertical takeoff, vertical landing rocket to demonstrate the performance of Select-DPC on an open-loop unstable nonlinear system. We show that Select-DPC successfully copes with the nonlinear dynamics and manages to stabilize the system at a given setpoint. A custom environment for the gymnasium suite Kwiatkowski et al. (2024) was used as a simulation platform. For a detailed description of the environment, we refer to Cummins et al. (2024).

The right column in Figure 1 shows the simulator setup and specifically, in the bottom figure, three closed-loop rocket trajectories. One for Select-DPC, one for standard DeePC using the full data set and one for Time-Windowed DeePC respectively. We use the Isomap-based selection method and  $N_{\text{cols}} = 500$ . While both the standard DeePC and the Time-Windowed DeePC controllers diverge, Select-DPC manages to regulate the system to the desired setpoint and stabilize it there.

Furthermore, we compare the performance of Select-DPC using the two presented selection methods on two data sets. Both data sets were collected using input sequences of the form  $u_t = a \cdot u_{t-1} + b \cdot n$ , with  $n \sim \mathcal{U}(-1, 1)$ . The data set *IID Data* uses  $(a, b) = (0, 1)$  and *Random Walk Data* uses  $(a, b) = (1, 0.1)$ . To ensure that a large region of the trajectory space was covered by the collected data, 100 simulations were run with randomized initial conditions and a maximum of 100 timesteps each, resulting in data sets with approximately 5000 data points.

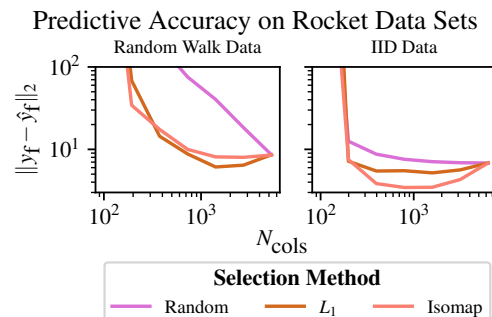


Figure 3: Residual between the ground-truth and least-squares predictions as a function of  $N_{\text{cols}}$  used in the predictor. Rightmost data point corresponds to standard DeePC.

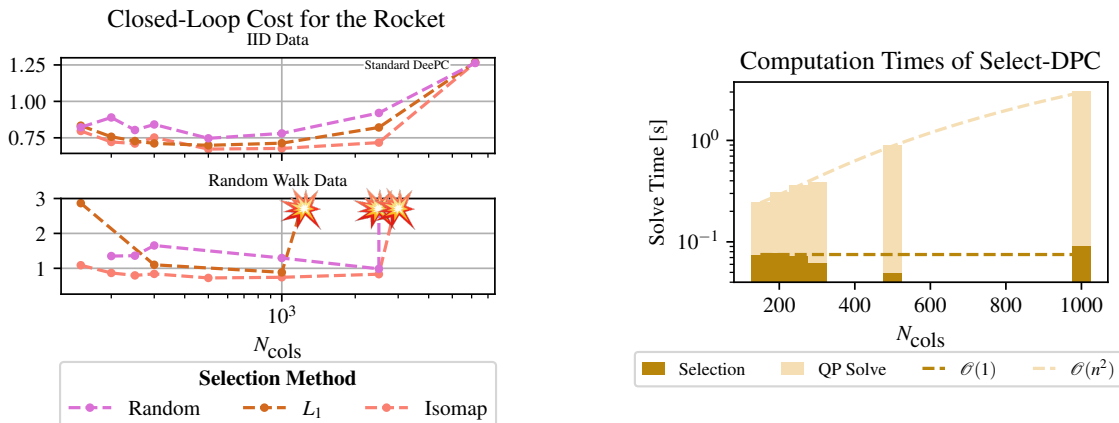


Figure 4: (*left*) Closed-loop cost as a function of data subset cardinality on two different data sets. Instances where the selection methods destabilized the system (i.e. infinite cost) have been omitted and the remaining finite cost data points are shown. (*right*) Solve times of Select-DPC as a function of  $N_{\text{cols}}$  decomposed into data selection and QP solve time. DeePC is used to solve the QP subproblem.

**Ablation Study** The comparisons of Select-DPC on the two data sets are two-fold. First, the resulting predictive accuracy of the selection methods are compared. Furthermore, a comparison of the realized closed-loop cost is given as a function of selection method and data set. As a benchmark, random sketching of the full data set at each decision moment is used.

The predictive accuracy of the methods was compared by computing the least squares solution of the predictor used in the DPC subproblem as a function of the number  $N_{\text{cols}}$  of trajectories in the selected set of data. For each trajectory in the holdout data set, constructed from trajectory-segments of a closed-loop trajectory, the  $N_{\text{cols}}$  closest trajectories (in spatial  $L_1$ -norm or using Isomap-embedding) under the given selection method were selected and the prediction of future measurements  $\hat{y}_f$ , given the corresponding future input sequence  $u_f$  were computed using the least square predictor of  $Hg = [u; y]$ . The plots in Figure 3 show the cumulative residuals between the predicted future trajectory and the ground truth realization  $\|\hat{y}_f - y_f\|_2$  over the entire validation set. In the extreme case where  $N_{\text{cols}} = \text{card}(\mathcal{D})$ , each of the methods coincides with the least squares solution of the predictor in standard DeePC. We observe that both methods of data selection (based on spatial  $L_1$ -norm or Isomap-embedding) lead to lower cumulative residual on the holdout validation set when compared to using the full data set or random sketching. In particular, there is a sweet spot in the number  $N_{\text{cols}}$  of trajectories to use in the predictor.

The left plot of Figure 4 shows the closed-loop performance of Select-DPC on the setpoint tracking task depicted in the right column of Figure 1 on both IID Data and Random Walk Data. We again observe on both data sets that selecting data outperforms using the full data set (i.e. standard DeePC) in terms of closed-loop performance, as a lower cost could be achieved. Using IID Data, the reduction in closed-loop cost is 1.9. Standard DeePC, using the full Random Walk data set, failed to stabilize the rocket.

The level of nonlinearity in the data has a significant effect on the performance of standard DeePC. Due to the high inertia of the rocket, the IID Data predominantly exhibits behaviors that can be represented reasonably well by a linear model, and hence, standard DeePC can be applied. However, this comes at a performance penalty as nonlinear effects are not accounted for. We observe that Select-DPC outperforms standard DeePC on IID Data and that data selection can still lead to

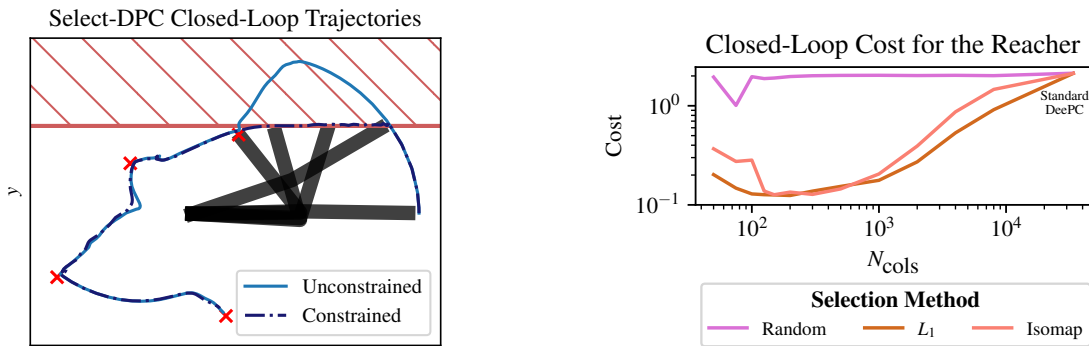


Figure 5: (left) Closed-loop trajectories for a constrained (dark blue) and unconstrained (blue) DPC subproblem. (right) Comparing the closed-loop cost incurred as a function of subset cardinality. Clearly there is a sweet spot in how many trajectories should be selected for the predictions.

performance increases if the data set does not fully capture the nonlinear behavior of the underlying system. The random walk structure of the input sequence in Random Walk Data, on the other hand, ensures that the nonlinearities of the system are sufficiently excited. This makes standard DeePC fail but provides Select-DPC with more global information about the nonlinear system.

**Computational Cost** The right plot of Figure 4 shows the computation times of Select-DPC (without LQ decomposition) as a function of number of subselected trajectories. The dominant factor in the solve times is solving the quadratic program, which empirically scales at least quadratically with  $N_{\text{cols}}$ , while data selection is a constant time operation with respect to  $N_{\text{cols}}$ . Concluding from Figure 4, reducing the number of trajectories is beneficial from both a computational and a closed-loop performance standpoint. Select-DPC not only increases performance in the nonlinear domain but also reduces computational cost compared to standard DeePC by one order of magnitude.

## 5.2. Planar Robotic Manipulator

We use the *Reacher* environment Kwiatkowski et al. (2024), which simulates a planar robotic manipulator, to demonstrate the ability of Select-DPC to control a highly nonlinear system and successfully track reference setpoints. Furthermore, we show that Select-DPC is effective in coping with and respecting a posteriori specified output constraints. The kinematic chain of the manipulator consists of two rigid bodies that are connected by revolute joints which are each actuated by a torque, giving it an action space of  $u \in \mathbb{R}^2$ . The measurement vector consists of the sine and cosine of both joint angles respectively, and the end-effector position in  $\mathbb{R}^2$ . Furthermore, we are also able to measure the angular velocity of each link, resulting in an observation space  $y \in \mathbb{R}^8$ . This system is more challenging to control using DeePC (or any linear control method) since the end-effector position is dictated by the composition of two rotations. In this environment, it is not unreasonable to assume that the angles must exceed  $90^\circ$  in order to reach a target. Furthermore, for a given end-effector position, the corresponding joint configuration is not necessarily unique.

Select-DPC was successfully able to track an end-effector reference as shown in the left column of Figure 1. We use  $L_1$ -based data selection and  $N_{\text{cols}} = 100$ . While Select-DPC is able to converge to the desired setpoint, standard DeePC fails to track the reference during the entire simulation horizon. Clearly, the open-loop predictions do not align with the closed-loop behavior, indicating that using the full data set results in bad predictive accuracy. Time-Windowed DeePC also fails to converge to the setpoint. We again investigate the effect of number of selected trajectories  $N_{\text{cols}}$ , as well as selection strategy on the resulting closed loop cost. For the Reacher, data were collected

using IID inputs. In total, 200 simulations were run with 200 steps each. Here, only IID inputs were collected instead of random walk because the system has lower inertia compared to the rocket and the nonlinearities could be excited more easily.

**Ablation Study** The right plot of Figure 5 shows the incurred closed-loop cost as a function of number of trajectories used in the implicit predictor. Similarly to the rocket, we observe a decrease in predictive accuracy and an increase in cost as the number of data points used in the predictor increases. While random sketching showed decent results in the rocket simulation, it fails completely in the reacher simulation as no accurate predictions could be generated, resulting in the controller not making any progress towards the setpoint and hence accumulating high closed-loop cost.

**Zero-Shot Generalizability of Select-DPC** The left plot of Figure 5 shows two closed-loop trajectories of Select-DPC where the controller was tasked with tracking a set of consecutive end-effector references. One simulation was left unconstrained, while a constraint on  $y$  was added to the DPC subproblem of the other one, indicated by the red region. Since the DeePC formulation used in Select-DPC admits arbitrary, a priori unknown output-constraints, we observe that Select-DPC is able to generalize to different tasks across the entire trajectory manifold of the Reacher by reusing the same data set and without structural changes to the controller. The structure of DeePC effectively decouples the dynamics learning/training process from the controller design and Select-DPC extends this to the nonlinear case. Indeed, both the constrained and unconstrained tasks use the same data set which is unaware of the structure of the predictive control problem. Thanks to the DeePC subproblem, this allows us to easily adapt the cost function or constraint terms for the task at hand in zero-shot fashion without requiring task specific data or additional online learning.

### 5.3. Cart-Pole inverted pendulum swing-up

We show that Select-DPC is capable of performing a cart-pole inverted pendulum swing-up. The input  $u \in \mathbb{R}$  is a force acting on the cart along the  $x$ -axis. Furthermore, the measurement  $y \in \mathbb{R}^5$  consists of the cart position, the sine and cosine of the angle between the pendulum and the vertical axis, as well as linear and angular velocities of the cart and pendulum respectively.

Due to the unstable nature of the setpoint and the system starting in a stable equilibrium position, data was gathered in closed-loop. Specifically, the data set consists of 200 demonstrations of successful but suboptimal swing-ups, collected using a stabilizing controller [Chatterjee et al. \(2002\)](#).

The left plot of Figure 6 shows the closed-loop trajectories of both Select-DPC and standard DeePC when tasked with tracking a setpoint at the upward facing, unstable equilibrium while starting from the downward facing stable equilibrium. While standard DeePC is unable to perform the swing-up and oscillates around the stable equilibrium, Select-DPC is successfully able to learn from the demonstrations in the data set and performs a swing-up to the desired vertical position and subsequently stabilizes the pendulum there. It is worthwhile to point out that the controller does not simply select the top trajectory in the selected data set. Instead, Select-DPC linearly combines several trajectory segments at each iteration of the solver to produce the swing-up (c.f. right plot of Figure 6). This is also evident from the fact that the horizon in the predictive controller is shorter than the episode length, thus the solver never has access to a full swing-up trajectory.

## 6. Limitations

Controllers relying on online optimization inherently require increased computational time compared to other methods. The current computational cost of Select-DPC of 200-400 ms prevents it from being employed in highly dynamic applications. A further bottleneck is Select-DPC’s memory footprint. Large prediction horizons and large data sets could violate memory constraints. A limiting

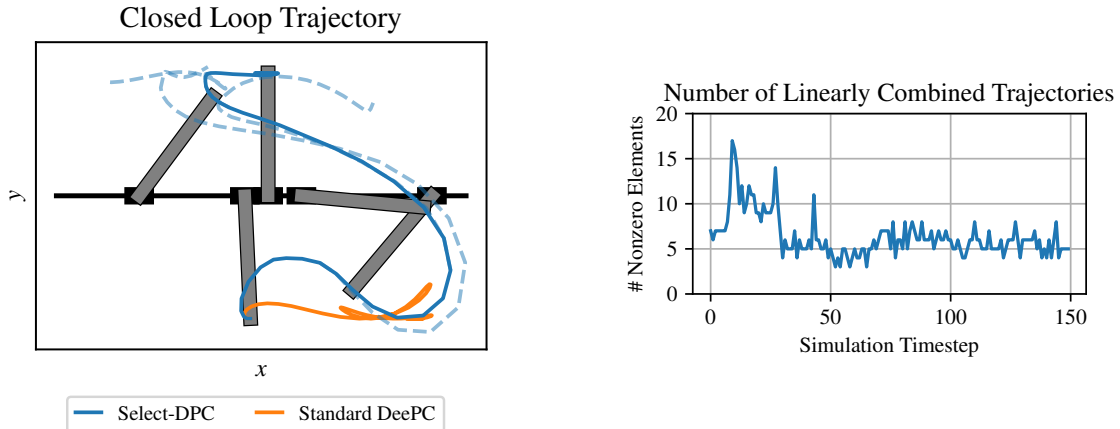


Figure 6: (left) Closed-loop trajectory for an inverted pendulum swing-up. Select-DPC (blue) is successful in transitioning from the downward facing stable equilibrium to the upward facing unstable equilibrium. Standard DeePC (orange) fails to swing up. The dashed lines indicate the open-loop prediction of Select-DPC at the four snapshot time instances,  $t=15, 22, 32$  and  $120$ . (right) Number of trajectories used to construct the final prediction at each sampling instant of Select-DPC.

factor of Select-DPC, hindering a study of its performance on a wide range of control tasks, is the necessity of hyper parameter tuning to achieve satisfactory control performance. While techniques exist for automatic controller tuning in this context, they have not been applied in this study and a large-scale study of Select-DPC to empirically quantify its capabilities is subject to future work.

## 7. Conclusion

This paper introduces Select-DPC, a novel approach to constraint-aware Data-driven Predictive Control for nonlinear systems. At each time instance, the method pre-processes the data set to determine the most relevant data, then passes the most relevant data to a convex optimization which determines the optimal control for the future horizon. The optimal control for the given timestep is implemented, and the algorithm progresses in a receding horizon fashion.

In contrast to other DPC methods, Select-DPC can be implemented easily, scales favorably with the number of collected data-points and increases performance outside of the linear domain. Furthermore, compared to sampling-based MPC, Select-DPC can explicitly handle output constraints.

We validated this on a set of three simulation environments. In the rocket simulator environment, we showed that for one of the data sets considered, Select-DPC cut closed-loop cost by a factor of 2. For the other data set, our method resulted in stable closed-loop behavior, where previous methods crashed. In the second set of simulations in the Reacher environment, we enabled zero-shot constrained setpoint tracking of a reference signal in a robotic reacher simulation. A task unachievable by previous DPC formulations. Finally, in the last set of simulations of a cart-pole inverted pendulum, we demonstrated successful transitioning between two different equilibria of the system by performing a pendulum swing-up.

A direction of future work is to investigate different selection methods, such as structure-exploiting selection methods or selection methods based on an information criterion. Another direction is to replace Select-DPC’s large training data set with a data-driven forward simulator of the dynamics (i.e., a “world model”), resulting in a constraint-aware alternative to sampling-based MPC.

## Acknowledgments

The authors would like to thank Xian Li for providing the simulation environment and data collection procedure for the inverted pendulum.

## References

- C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Jan Van Den Bussche, and Victor Vianu, editors, *Database Theory — ICDT 2001*, volume 1973, pages 420–434. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. Series Title: Lecture Notes in Computer Science.
- T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006. doi: 10.1109/MRA.2006.1678144.
- J. Berberich and F. Allgöwer. An Overview of Systems-Theoretic Guarantees in Data-Driven Model Predictive Control. October 2024. Publisher: Annual Reviews.
- J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer. Linear tracking MPC for nonlinear systems Part II: The data-driven case. *IEEE Transactions on Automatic Control*, 67(9):4406–4421, 2022.
- F. Berkenkamp, A. P. Schoellig, and A. Krause. Safe Controller Optimization for Quadrotors with Gaussian Processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 491–496, 2016.
- P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. doi: 10.1109/34.121791.
- Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, and P. L’Ecuyer. Chapter 3 - The Cross-Entropy Method for Optimization. In C. R. Rao and Venu Govindaraju, editors, *Handbook of Statistics*, volume 31 of *Handbook of Statistics*, pages 35–59. Elsevier, January 2013.
- V. Breschi, A. Chiuso, and S. Formentin. Data-driven predictive control in a stochastic setting: a unified framework. *Automatica*, 152:110961, June 2023.
- D. Chatterjee, A. Patra, and H. K. Joglekar. Swing-up and stabilization of a cart–pendulum system under restricted cart track length. *Systems & Control Letters*, 47(4):355–364, 2002.
- J. Coulson, J. Lygeros, and F. Dörfler. Data-Enabled Predictive Control: In the Shallows of the DeePC. In *2019 18th European Control Conference (ECC)*, pages 307–312, 2019.
- M. Cummins, A. Padoan, K. Moffat, F. Dorfler, and J. Lygeros. DeePC-Hunt: Data-enabled Predictive Control Hyperparameter Tuning via Differentiable Optimization, December 2024. arXiv:2412.06481 [math].
- M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577–585, 2002.

- F. Dörfler, J. Coulson, and I. Markovskiy. Bridging Direct and Indirect Data-Driven Control Formulations via Regularizations and Relaxations. *IEEE Transactions on Automatic Control*, 68(2): 883–897, February 2023. Conference Name: IEEE Transactions on Automatic Control.
- E. Elokda, J. Coulson, P. N. Beuchat, J. Lygeros, and F. Dörfler. Data-enabled predictive control for quadcopters. *International Journal of Robust and Nonlinear Control*, 31(18):8916–8936, 2021.
- C. Fefferman, S. Mitter, and H. Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, October 2016.
- L. Huang, J. Lygeros, and F. Dörfler. Robust and Kernelized Data-Enabled Predictive Control for Nonlinear Systems. *IEEE Transactions on Control Systems Technology*, 32(2):611–624, March 2024. Conference Name: IEEE Transactions on Control Systems Technology.
- W. Jia, M. Sun, J. Lian, and S. Hou. Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, 8(3):2663–2693, 2022.
- T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel. Model-Ensemble Trust-Region Policy Optimization, October 2018. arXiv:1802.10592 [cs].
- A. Kwiatkowski, M. Towers, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, H. Tan, and O. G. Younis. Gymnasium: A Standard Interface for Reinforcement Learning Environments, October 2024. arXiv:2407.17032.
- M. Lazar. Basis functions nonlinear data-enabled predictive control: Consistent and computationally efficient formulations, November 2023. arXiv:2311.05360 [cs, eess, math].
- J. A. Lee and M. Verleysen, editors. *Nonlinear Dimensionality Reduction*. Information Science and Statistics. Springer, New York, NY, 2007. ISBN 978-0-387-39350-6.
- Y. Lian and C. N. Jones. Nonlinear data-enabled prediction and control. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 523–534. PMLR, 07 – 08 June 2021.
- I. Markovskiy, L. Huang, and F. Dörfler. Data-Driven Control Based on the Behavioral Approach: From Theory to Applications in Power Systems. *IEEE Control Systems*, 43(5):28–68, October 2023.
- A. Martinelli, M. Gargiani, M. Draskovic, and J. Lygeros. Data-Driven Optimal Control of Affine Systems: A Linear Programming Perspective. *IEEE Control Systems Letters*, 6:3092–3097, 2022. Conference Name: IEEE Control Systems Letters.
- T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker. Model-based Reinforcement Learning: A Survey, March 2022. arXiv:2006.16712 [cs].
- M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, May 1999.

- A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine. Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566, May 2018.
- R. S Sutton and A. G Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- J. B. Tenenbaum, V. De Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000.
- J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, April 2005.
- J.C. Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control*, 36(3):259–294, 1991.
- G. Williams, A. Aldrich, and E. A. Theodorou. Model Predictive Path Integral Control: From Theory to Parallel Computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, February 2017.

## Appendix A. DeePC

Data-enabled Predictive Control (DeePC) solves a receding horizon optimal control problem based purely on data for a linear time-invariant system (LTI) of the form

$$y_f(k) = F_p \begin{bmatrix} u_p(k-1) \\ y_p(k-1) \end{bmatrix} + F_f u_f(k), \quad (2)$$

where  $u$  and  $y$  are partitioned into “past” and “future” quantities  $u_p, u_f$  and  $y_p, y_f$  respectively. This means that the sequence of predicted future measurements, the past measurements used by the predictor, the sequence of past inputs applied to the system and the future control input sequence respectively given by

$$\begin{aligned} y_f(k) &:= [y(k)^\top \ \dots \ y(k+T_f-1)^\top]^\top \in \mathbb{R}^{T_f \cdot p}, \\ u_f(k) &:= [u(k)^\top \ \dots \ u(k+T_f-1)^\top]^\top \in \mathbb{R}^{T_f \cdot m}, \\ y_p(k-1) &:= [y(k-T_p)^\top \ \dots \ y(k-1)^\top]^\top \in \mathbb{R}^{T_p \cdot p}, \text{ and} \\ u_p(k-1) &:= [u(k-T_p)^\top \ \dots \ u(k-1)^\top]^\top \in \mathbb{R}^{T_p \cdot m}. \end{aligned}$$

Let  $\mathcal{U}$  and  $\mathcal{Y}$  denote constraint sets on the input and output  $u_f$  and  $y_f$  respectively. Given a set of stage cost functions  $c_i$  and past trajectory of the system, we are interested in resolving the following problem in a receding horizon fashion.

$$\min_{u_f, y_f} \sum_{i=0}^{T_f-1} c_i(u_{f,i}, y_{f,i}) \quad (3a)$$

$$\text{s.t. } y_f = F_p \begin{bmatrix} u_p \\ y_p \end{bmatrix} + F_f u_f, \quad (3b)$$

$$(u_f, y_f) \in \mathcal{U} \times \mathcal{Y}. \quad (3c)$$

Without access to the model (2), we will need to reconstruct it from offline data. Let  $v_T = \{v(i)\}_{i=0}^{T-1}$  denote measurements of a signal  $v$  with length  $T \in \mathbb{Z}_{\geq 1}$ . We define the Hankel matrix of depth  $L$  of  $v_T$  as

$$H_L(v_T) = \begin{bmatrix} v(0) & v(1) & \dots & v(T-L-1) \\ v(1) & v(2) & \dots & v(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ v(L-1) & v(L) & \dots & v(T-1) \end{bmatrix} \quad (4)$$

Note that each column of the Hankel matrix contains a trajectory of length  $L$ .

**Definition 1 (Persistency of Excitation)** *For a given sequence  $v_T$ , we call the sequence persistently exciting of order  $L$  if  $H_L(v_T)$  has full row rank.*

Using persistently exciting inputs, Willems’ Fundamental Lemma allows us to parametrize all finite-length trajectories of a controllable linear system.

**Lemma 2 (The Fundamental Lemma [Willems et al. \(2005\)](#))** *Consider a controllable linear time-invariant system of order  $n$ . Given an input sequence  $u_T$  which is persistently exciting of order  $L \cdot m + n$  and the corresponding output sequence  $y_T$ , then there exists  $g$  such that any admissible trajectory  $(u, y)$  of length  $L$  can be expressed as a linear combination*

$$\begin{bmatrix} H_L(u_T) \\ H_L(y_T) \end{bmatrix} g = \begin{bmatrix} u \\ y \end{bmatrix}. \quad (5)$$

DeePC is based on the use of this implicit predictor (5) as a proxy for the explicit model representation in (2). In an online manner, DeePC matches the “past” data with the most recently seen inputs and outputs, and uses the “future” data to match a prediction over which we optimize. This results in a receding-horizon predictive controller based purely on data:

$$\min_{u_f, y_f, g} \sum_{i=0}^{T_f-1} c_i(u_{f,i}, y_{f,i}) + r(g) \quad (6a)$$

$$\text{s.t.} \quad \begin{bmatrix} H_{T_p+T_f}(u_T) \\ H_{T_p+T_f}(y_T) \end{bmatrix} g = \begin{bmatrix} u_p \\ u_f \\ y_p \\ y_f \end{bmatrix}, \quad (6b)$$

$$(u_f, y_f) \in \mathcal{U} \times \mathcal{Y}. \quad (6c)$$

Here,  $r(g)$  denotes suitable regularizers on  $g$  [Dörfler et al. \(2023\)](#) which improve robustness of the controller in the presence of noise. The implicit predictor in (5) can readily be extended to affine systems as follows [Martinelli et al. \(2022\)](#); [Berberich et al. \(2022\)](#)

$$\begin{bmatrix} H_t(u_T) \\ H_t(y_T) \\ \mathbb{1}^\top \end{bmatrix} g = \begin{bmatrix} u \\ y \\ 1 \end{bmatrix}. \quad (7)$$

### A.1. From Implicit to Explicit Predictor

Partitioning the predictor (7) into past and future states results in

$$\begin{bmatrix} U_p \\ U_f \\ Y_p \\ Y_f \\ \mathbb{1}^\top \end{bmatrix} g = \begin{bmatrix} u_p \\ u_f \\ y_p \\ y_f \\ 1 \end{bmatrix}, \quad (8)$$

which is an implicit predictor for the future measurement trajectory  $y_f$  given past measurements  $y_p$ , the corresponding past input sequence  $u_p$  and a future input sequence  $u_f$ . Using the partitions

$$H_z := \begin{bmatrix} U_p \\ U_f \\ Y_p \\ \mathbb{1}^\top \end{bmatrix}, \quad z := \begin{bmatrix} u_p \\ u_f \\ y_p \\ 1 \end{bmatrix}, \quad (9)$$

we can eliminate  $g$  from the implicit predictor in Equation (8) and write the equivalent explicit predictor

$$y_f = Y_f H_z^\dagger z + Y_f N_{H_z} g_0, \quad (10)$$

where  $A^\dagger$  denotes the pseudo-inverse of  $A$ ,  $N_{H_z}$  is the null-space projection matrix of  $H_z$  and  $g_0$  is an arbitrary perturbation to the least squares solution  $g = H_z^\dagger z$ . Assuming  $H_z$  has full row rank, then the least squares solution (i.e.  $N_{H_z} g_0 = 0$ ) is

$$y_f = Y_f H_z^\top \left( H_z H_z^\top \right)^{-1} z. \quad (11)$$

## Appendix B. SQP-MPC

Nonlinear MPC tries to optimize a cost function subject to the state evolution of a nonlinear system of the form

$$y_f(k) = f(u_p(k-1), u_f(k), y_p(k-1)). \quad (12)$$

If we had access to  $f$ , then we could use the receding-horizon optimal control problem

$$\min_{u_f, y_f} \sum_{i=0}^{T_f-1} c_i(u_{f,i}, y_{f,i}) \quad (13a)$$

$$\text{s.t. } y_f = f(u_p, u_f, y_p), \quad (13b)$$

$$(u_f, y_f) \in \mathcal{U} \times \mathcal{Y} \quad (13c)$$

to obtain an optimal control input sequence  $u_f^*$ . One method of solving a nonlinear optimal control problem of this form is using SQP, which repeatedly solves a linearization of the nonlinear problem (13a), resulting in subproblem (14). The solution of said subproblem is then used to update the estimate of the optimal open-loop solution  $(\tilde{u}_f, \tilde{y}_f)$ .

$$\min_{\Delta u_f, \Delta y_f} \sum_{i=0}^{T_f-1} [\Delta y_{f,i}^\top \quad \Delta u_{f,i}^\top \quad 1] \tilde{H}_i \begin{bmatrix} \Delta y_{f,i} \\ \Delta u_{f,i} \\ 1 \end{bmatrix} \quad (14a)$$

$$\text{s.t. } \Delta y_f = F_p \begin{bmatrix} \Delta u_p \\ \Delta y_p \end{bmatrix} + F_f \Delta u_f, \quad (14b)$$

$$\Delta z_p = \begin{bmatrix} u_p(k) - \tilde{u}_p \\ y_p(k) - \tilde{y}_p \end{bmatrix}, \quad (14c)$$

$$(\Delta u_{f,i}, \Delta y_{f,i}) \in \mathcal{U} \times \mathcal{Y} \quad \forall i \in \{0, \dots, T_f - 1\}, \quad (14d)$$

where the symbols  $F_p := \frac{\partial}{\partial [u_p^\top \quad y_p^\top]^\top} f(u_p, y_p, \tilde{u}_f)$  and  $F_f := \frac{\partial}{\partial \tilde{u}_f} f(u_p, y_p, \tilde{u}_f)$  indicate the respective Jacobians of system (12) evaluated at the current solution estimate  $(u_p, y_p, \tilde{u}_f, \tilde{y}_f)$ .

The full SQP algorithm can be summarized as follows [Diehl et al. \(2002\)](#)

---

### Algorithm 4 SQP-MPC (IO Representation)

---

- 1: **function** SQP-MPC( $z_p$ )
  - 2:   **while** not converged **do**
  - 3:     linearize dynamics, constraints and cost at  $(z_p, \tilde{u}_f, \tilde{y}_f)$
  - 4:     Solve Equation (14)
  - 5:      $(\tilde{u}_f, \tilde{y}_f) \leftarrow (\tilde{u}_f, \tilde{y}_f) + (\Delta u_f, \Delta y_f)$
  - 6:   **end while**
  - 7:   Return  $\tilde{u}_{f,0}$
  - 8: **end function**
- 

#### B.1. Equivalence between Select-DPC and SQP-MPC

Select-DPC has a very natural interpretation as solving a nonlinear MPC problem using SQP, where the analytical Jacobian computation is replaced by a data-driven approach of estimating the Jacobian

of the input-output behavior from data. Specifically, one can select data points “close” to the current linearization point (given by the open loop solution), resulting in an equivalent, data-driven approximation of  $F_p$  and  $F_f$  in Equation (14) that approaches the analytical Jacobians assuming sufficient amounts of data in the neighborhood around the linearization point (i.e.  $\text{card}(\mathcal{D})$  tending to infinity).

## Appendix C. Isomap

Isomap, short for Isometric Mapping, is a nonlinear dimensionality reduction technique introduced by Tenenbaum [Tenenbaum et al. \(2000\)](#). It can be seen as an extension of classical Multidimensional Scaling (MDS) where, instead of using Euclidean interpoint distances, geodesic distances between the data points are preserved. These geodesic distances are estimated using shortest distances on a connected graph which connects neighboring data points. These shortest paths are then arranged in a distance matrix  $D$ . Isomap then finds an embedding by applying classical MDS. Computing the distance of a new query data point to the data set in the embedding dimension is done by first linking the new data point into the neighborhood graph and then projecting the data point into the embedding space.

Isomap is able to capture the overall nonlinear nature of the data manifold and is guaranteed to produce a global optimizer of its reconstruction loss. Furthermore, due to preserving geodesic distances, points close in the input space are also close in the embedding space, resulting in interpretable results. This is especially relevant for Select-DPC which is build upon the idea of selecting data points according to their associated distance to a query point.

The relative contrast of a data set  $\Delta = (d_{\max} - d_{\min})/d_{\min}$ , with  $d_{\max/\min}$  indicating the maximum and minimum norm values of the data set respectively, is a helpful surrogate to determine how distinct each data point appears under a given distance metric. Figure 7 shows the relative contrast of a data set collected from the rocket simulator in Section 5 as a function of trajectory dimensionality when embedded using Isomap. It compares relative contrast given a fixed embedding dimensionality as well as a varying one, which increases according to  $n + T_f \cdot m$ , to the relative contrast of the  $L_1$  norm. Both Isomap methods have better relative contrast compared to  $L_1$  selection. While a varying embedding dimensionality offers greater relative contrast for small trajectory lengths, we encounter diminishing returns for large trajectory lengths.

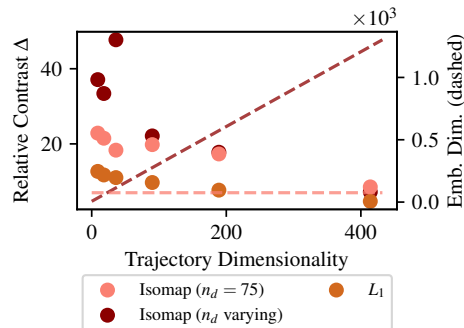


Figure 7: Relative contrast with Isomap embedding compared to  $L_1$  selection.

### C.1. Isomap Parameter Selection

Isomap comes with a set of hyper parameters that need to be selected for the data set at hand. Specifically, the number of closest data points that should be considered as neighbors during the neighborhood graph construction and the dimensionality of the embedding space. Figure 8 shows the reconstruction error of the data sets as a function of number of graph neighbors used during adjacency graph construction and number of dimensions of the embedding space. The graphs show a minimum around 10 graph neighbors while we can observe diminishing returns from increasing the embedding dimensionality from 64 to 128 which is consistent with the formula  $T_f \cdot m + n$  which suggests that the locally linear system has a dimensionality of 96.

Furthermore, note the sharp increase in reconstruction error as the number of graph neighbors increases after the local minimum. This demonstrates the effect of bad hyperparameter selection

which can lead to short-circuiting or over-connectedness in the graph, leading to worse embedding performance as the local manifold structure is lost [Tenenbaum et al. \(2000\)](#).

Isomap Embedding Error

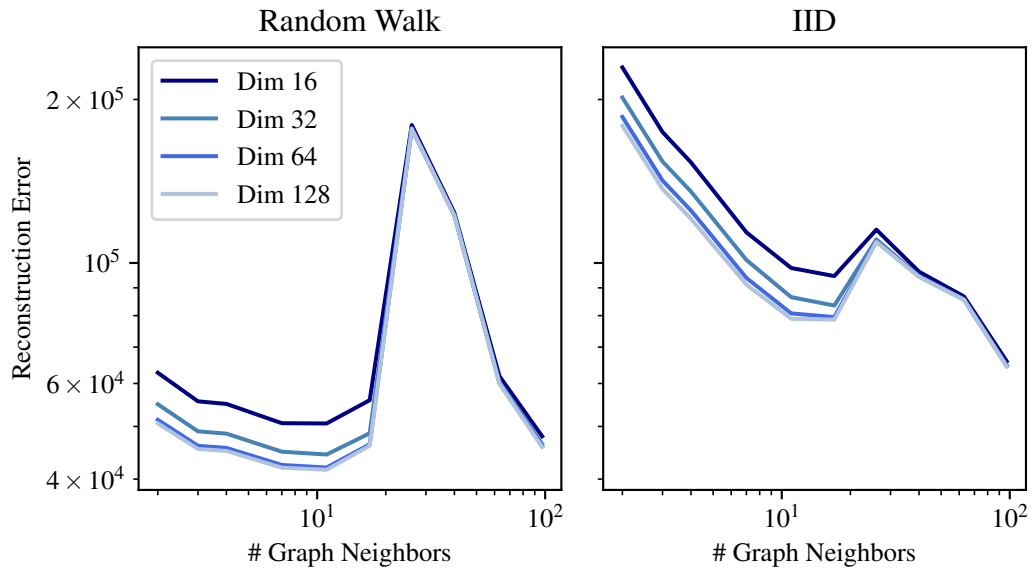


Figure 8: Isomap reconstruction error as a function a embedding dimensionality and number of neighbors used in the adjacency graph.

## Appendix D. Experiment Hyperparameters

All experiments were performed using a quadratic tracking cost of the form

$$c_i(u_{f,i}, y_{f,i}) = \|y_{f,i} - y_r\|_Q + \|u_{f,i}\|_R, \quad (15)$$

where  $Q$  and  $R$  are positive (semi-)definite cost matrices and  $y_r$  is a reference setpoint. Furthermore, for the regularization term  $r(g)$  we use 1-norm and projection regularization [Markovsky et al. \(2023\)](#) with weights  $\lambda_1$  and  $\lambda_\Pi$  respectively. The following tables shows the hyperparameters for the three sets of simulations.

Parameter	Value
convergence criterion $n_{\max}$	1
$Q$	$\text{diag}([40 \ 20 \ 20 \ 1 \ 3000 \ 30])$
$R$	$\text{diag}([10 \ 10 \ 10])$
$\lambda_1$	0
$\lambda_\pi$	5000

Table 1: Hyperparameters for the rocket simulation environment.

Parameter	Value
convergence criterion $n_{\max}$	5
$Q$	$\text{diag}([0 \ 0 \ 0 \ 0 \ 40\,000 \ 40\,000 \ 10 \ 10])$
$R$	$\text{diag}([10 \ 10])$
$\lambda_1$	10
$\lambda_\pi$	10 000

Table 2: Hyperparameters for the reacher simulation environment.

Parameter	Value
convergence criterion $n_{\max}$	1
$Q$	$\text{diag}([5 \ 0 \ 10\,000 \ 0.1 \ 0.1])$
$R$	1
$\lambda_1$	50 000
$\lambda_\pi$	0

Table 3: Hyperparameters for the cart-pole simulation environment.