Fragmented Data-Driven Predictive Control: Towards smaller datasets ⋆

Ruslan Shaiakhmetov * **Danilo Pianini** * **Valter Venusti** ** **Alessandro V. Papadopoulos** ***

* *Alma Mater Studiorum—Università di Bologna, Italy*
** *Dallara Automobili S.p.A., Italy*
*** *Mälardalen University, Sweden*

Abstract: Data-driven control techniques, such as DeePC, enable controller synthesis directly from data, but remain challenging for nonlinear or stochastic systems with limited data. Existing extensions, such as S-DeePC, address this through problem decomposition, but rely on indirect formulations. This paper presents F-DeePC, a fragmented DeePC approach that builds the constraint matrices directly and splits the prediction horizon into shorter fragments. The construction supports heterogeneous fragment lengths, while this paper focuses on the homogeneous case. The resulting formulation improves flexibility and data efficiency, yielding better tracking performance under limited data and nonlinear dynamics.

Keywords: Data-driven control, Predictive control, Optimal control.

1. INTRODUCTION

Data-driven control has emerged as a novel paradigm for control system design (Coulson et al., 2019), providing an alternative to model-based approaches such as Model Predictive Control (MPC) when first-principles models are unavailable, yet input-output data are accessible.

Existing data-driven methods are commonly divided into indirect and direct approaches (Krishnan and Pasqualetti, 2021; Dörfler et al., 2023). Indirect methods first identify a model from data and then design a controller using classical tools, but their performance depends on the accuracy of the identified model. Direct methods instead compute control laws directly from data. Among them, Data-enabled Predictive Control (DeePC) (Coulson et al., 2019) builds on Willems' fundamental lemma (Willems et al., 2005; Rapisarda and Pal, 2023) and has been applied, for instance, to power systems (Huang et al., 2022; Markovsky et al., 2023) and aerial robotics (Elokda et al., 2021; Bouffard et al., 2012).

Despite its success, applying DeePC to stochastic and nonlinear systems remains challenging. Existing remedies include constraint relaxation (Abolpour et al., 2024; Seuret et al., 2023), robust regularization (Dörfler et al., 2022, 2023), dataset selection or compression (Huang et al., 2024; Pillonetto et al., 2014; Alsalti et al., 2024), and uncertainty-aware regularization (Chiuso et al., 2025). A complementary strategy is to replace one long trajectory representation with several shorter ones, which can improve numerical conditioning under noisy data (van Waarde et al., 2020).

Building on this idea, Segmented Data-enabled Predictive Control (S-DeePC) (O'Dwyer et al., 2023) partitions the prediction problem into shorter segments, improving tracking and robustness in data-limited noisy settings. However, S-DeePC relies on an indirect multistep predictor and uses a fixed segment structure. This paper introduces Fragmented Data-enabled Predictive Control (F-DeePC), which retains horizon splitting but formulates each fragment through direct DeePC-type behavioral constraints inside a single Quadratic Programming (QP).

2. BACKGROUND

Data-enabled Predictive Control (DeePC). DeePC is a predictive control scheme that predicts future outputs and selects inputs directly from past input-output data. The dataset contains N trajectories of length T, with stacked inputs $\mathbf{u} \in \mathbb{R}^{mT}$ and outputs $\mathbf{y} \in \mathbb{R}^{pT}$, arranged column-wise for Multiple-Input Multiple-Output (MIMO) systems. Each trajectory is split into past ($\mathbf{u}_{\triangleleft} \in \mathbb{R}^{mT_{\triangleleft}}$, $\mathbf{y}_{\triangleleft} \in \mathbb{R}^{pT_{\triangleleft}}$) and future ($\mathbf{u}_{\triangleright} \in \mathbb{R}^{mT_{\triangleright}}$, $\mathbf{y}_{\triangleright} \in \mathbb{R}^{pT_{\triangleright}}$) parts, where $T_{\triangleleft}$ is the initial window and $T_{\triangleright}$ the prediction horizon. Following Markovsky and Rapisarda (2008), DeePC imposes

$$\underbrace{\begin{pmatrix} \mathbf{u}_{\triangleleft}^1 & \mathbf{u}_{\triangleleft}^2 & \cdots & \mathbf{u}_{\triangleleft}^N \\ \mathbf{y}_{\triangleleft}^1 & \mathbf{y}_{\triangleleft}^2 & \cdots & \mathbf{y}_{\triangleleft}^N \\ \mathbf{u}_{\triangleright}^1 & \mathbf{u}_{\triangleright}^2 & \cdots & \mathbf{u}_{\triangleright}^N \\ \mathbf{y}_{\triangleright}^1 & \mathbf{y}_{\triangleright}^2 & \cdots & \mathbf{y}_{\triangleright}^N \end{pmatrix}}_{\mathbf{H} \in \mathbb{R}^{T(m+p) \times N}} \underbrace{\begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{pmatrix}}_{\mathbf{g} \in \mathbb{R}^N} = \underbrace{\begin{pmatrix} \mathbf{u}_{\triangleleft}^{ini} \\ \mathbf{y}_{\triangleleft}^{ini} \\ \mathbf{u}_{\triangleright} \\ \mathbf{y}_{\triangleright} \end{pmatrix}}_{\mathbf{L} \in \mathbb{R}^{T(m+p)}}. \tag{1}$$

Here, $\mathbf{g}$ are trajectory-combination coefficients, while $\mathbf{u}_{\triangleleft}^{ini} \in \mathbb{R}^{mT_{\triangleleft}}$ and $\mathbf{y}_{\triangleleft}^{ini} \in \mathbb{R}^{pT_{\triangleleft}}$ are the measured histories. Under the conditions of Willems' Fundamental Lemma (Markovsky and Rapisarda, 2008), any compatible

⋆ This work is supported by the project "Ottimizzazione computazione di un modello di simulazione dinamica del veicolo" funded by the European Union - NextGenerationEU and Dallara Automobili S.p.A. through the Italian "National Recovery and Resilience Plan" (PNRR) Mission 4, Component 2, Investment 3.3 (DM 352/2022) - CUP J33C22001400009 and the Swedish Research Council (VR), and by the Knowledge Foundation (KKS).

Linear Time-Invariant (LTI) trajectory lies in the span of the columns of $\mathbf{H}$.

A length-L time series yields $N = L-T+1$ subtrajectories through a sliding window of width T. For a scalar signal $h \in \mathbb{R}^L$, this gives

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & \cdots & h_{L-T+1} \\ h_2 & h_3 & \cdots & h_{L-T+2} \\ \vdots & \vdots & \ddots & \vdots \\ h_T & h_{T+1} & \cdots & h_L \end{bmatrix}. \tag{2}$$

Given (1), DeePC computes future inputs by solving

$$\begin{aligned} \underset{\mathbf{g},\mathbf{u}_\blacktriangleright,\mathbf{y}_\blacktriangleright}{\text{minimize}} \quad & V(\mathbf{g},\mathbf{u}_\blacktriangleright,\mathbf{y}_\blacktriangleright) \\ \text{subject to} \quad & \mathbf{H}\mathbf{g} = \mathbf{L},\ \mathbf{u}_\blacktriangleright \in \mathcal{U},\ \mathbf{y}_\blacktriangleright \in \mathcal{Y}, \end{aligned} \tag{3}$$

typically with

$$V(\mathbf{g},\mathbf{u}_\blacktriangleright,\mathbf{y}_\blacktriangleright) = \|\mathbf{y}_\blacktriangleright - \mathbf{r}\|_Q^2 + \|\mathbf{u}_\blacktriangleright\|_R^2, \tag{4}$$

where $Q \succeq 0$, $R \succ 0$, and $\mathbf{r}$ spans $T_\blacktriangleright$ steps. The first input is applied, the measured window is updated, and the procedure is repeated.

For noisy data or modest nonlinearities, the exact behavioral constraint is relaxed with output slacks $\boldsymbol{\varepsilon}_\triangleleft \in \mathbb{R}^{pT_\triangleleft}$ and $\boldsymbol{\varepsilon}_\blacktriangleright \in \mathbb{R}^{pT_\blacktriangleright}$, and an ℓ_1 penalty on $\mathbf{g}$ is commonly added to reduce overfitting (Coulson et al., 2019; Dörfler et al., 2023; Shafieezadeh-Abadeh et al., 2019). This yields

$$\begin{aligned} \underset{\mathbf{g},\mathbf{u}_\blacktriangleright,\mathbf{y}_\blacktriangleright,\boldsymbol{\varepsilon}_\blacktriangleright,\boldsymbol{\varepsilon}_\triangleleft}{\text{minimize}} \quad & V(\mathbf{g},\mathbf{u}_\blacktriangleright,\mathbf{y}_\blacktriangleright,\boldsymbol{\varepsilon}_\blacktriangleright,\boldsymbol{\varepsilon}_\triangleleft) + \\ & \lambda_g\|\mathbf{g}\|_1 + \lambda_\triangleleft\|\boldsymbol{\varepsilon}_\triangleleft\|_1 + \lambda_\blacktriangleright\|\boldsymbol{\varepsilon}_\blacktriangleright\|_1 \\ \text{subject to} \quad & \mathbf{u}_\blacktriangleright \in \mathcal{U},\ \mathbf{y}_\blacktriangleright \in \mathcal{Y}, \\ & \boldsymbol{\varepsilon}_\triangleleft \geqslant \mathbf{0},\ \boldsymbol{\varepsilon}_\blacktriangleright \geqslant \mathbf{0}, \\ & \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\varepsilon}_\triangleleft \\ \mathbf{0} \\ -\boldsymbol{\varepsilon}_\blacktriangleright \end{bmatrix} \leqslant \mathbf{H}\mathbf{g} - \mathbf{L} \leqslant \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\varepsilon}_\triangleleft \\ \mathbf{0} \\ \boldsymbol{\varepsilon}_\blacktriangleright \end{bmatrix}. \end{aligned} \tag{5}$$

Segmented DeePC. For long horizons, DeePC may be sensitive to unmeasured disturbances (Alpago et al., 2020; Berberich et al., 2021). S-DeePC mitigates this by splitting the prediction interval into F equal-length segments (O'Dwyer et al., 2023). Each segment has past and future windows of length $T_\triangleleft$, so the predictor is identified from many short windows rather than fewer long trajectories, improving conditioning under noisy data.

Unlike the direct relation (1), S-DeePC uses an *indirect* least-squares predictor. For a segment dataset

$$H_s = \begin{bmatrix} U_\triangleleft^s \\ Y_\triangleleft^s \\ U_\blacktriangleright^s \\ Y_\blacktriangleright^s \end{bmatrix},$$

the predictor is

$$P_s^* = Y_\blacktriangleright^s \begin{bmatrix} U_\triangleleft^s \\ Y_\triangleleft^s \\ U_\blacktriangleright^s \end{bmatrix}^\dagger = \begin{bmatrix} P_{u,\triangleleft}^s & P_{y,\triangleleft}^s & P_{u,\blacktriangleright}^s \end{bmatrix}. \tag{6}$$

For the first segment,

$$y_1 = P_{u,\triangleleft}^s \mathbf{u}_\triangleleft^{ini} + P_{y,\triangleleft}^s \mathbf{y}_\triangleleft^{ini} + P_{u,\blacktriangleright}^s u_1, \tag{7}$$

whereas subsequent segments use the previous prediction as their past window:

$$y_i = P_{u,\triangleleft}^s u_{i-1} + P_{y,\triangleleft}^s y_{i-1} + P_{u,\blacktriangleright}^s u_i, \qquad i = 2,\dots,F. \tag{8}$$

Equivalently,

$$\begin{aligned} \operatorname{col}(y_1,\dots,y_F) = {} & (I_F \otimes P_{u,\triangleleft}^s)\operatorname{col}(\mathbf{u}_\triangleleft^{ini}, u_1,\dots,u_{F-1}) \\ & + (I_F \otimes P_{y,\triangleleft}^s)\operatorname{col}(\mathbf{y}_\triangleleft^{ini}, y_1,\dots,y_{F-1}) \\ & + (I_F \otimes P_{u,\blacktriangleright}^s)\operatorname{col}(u_1,\dots,u_F). \end{aligned}$$

Thus, S-DeePC optimizes over segment inputs and outputs, not behavioral coefficients. A direct segmented formulation would instead impose $H_s\mathbf{g}_i = \operatorname{col}(u_{\triangleleft,i}, y_{\triangleleft,i}, u_i, y_i)$, which is the viewpoint adopted by F-DeePC, with fragment future lengths not tied to the initial-window length.

3. PROPOSED APPROACH: FRAGMENTED DEEPC

We generalize S-DeePC by retaining horizon splitting while returning to a direct behavioral formulation. Instead of identifying an indirect predictor and recursively propagating segment outputs through (8), F-DeePC builds a single QP with one DeePC-type behavioral constraint per fragment. The fragment future length is independent of the initial-window length, so the construction can accommodate heterogeneous horizons; below, we focus on the homogeneous case.

Fragmentation avoids representing the whole prediction horizon with one long data-driven trajectory. While standard DeePC uses one coefficient vector for the measured past and the full future input-output sequence, F-DeePC represents the prediction as a chain of shorter behavioral fragments. The first fragment is initialized from the measured past, and subsequent fragments use previous predictions as their past window. Inter-fragment consistency constraints make the concatenation behave as one full-horizon trajectory.

Thus, unlike S-DeePC, F-DeePC keeps the fragment coefficients $\mathbf{g}_i$ as decision variables instead of first identifying a least-squares predictor. The global variables $\mathbf{u}_\blacktriangleright$ and $\mathbf{y}_\blacktriangleright$ describe the full predicted trajectory, while each $\mathbf{g}_i$ certifies that the corresponding fragment is data-compatible.

The controller solves a quadratic program (QP)

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2}x^\top Q x + q^\top x \\ \text{subject to} \quad & b_l \leqslant Ax \leqslant b_u. \end{aligned} \tag{9}$$

The cost contains the tracking and input-effort terms of (4), plus linear penalties on nonnegative slack variables. An ℓ_1 penalty on the unrestricted coefficient vector is implemented through the standard positive/negative epigraph construction.

3.1 Fragment indexing and selection matrices

Let F be the number of fragments and let each fragment have past length $T_\triangleleft$ and future length $T_{\mathcal{F}\blacktriangleright}$. The full prediction horizon is $T_\blacktriangleright = FT_{\mathcal{F}\blacktriangleright}$, and one fragment has total length $T_\mathcal{F} := T_\triangleleft + T_{\mathcal{F}\blacktriangleright}$.

Let the fragment Hankel matrix be

$$H_\mathcal{F} := \begin{bmatrix} U_\triangleleft^\mathcal{F} \\ Y_\triangleleft^\mathcal{F} \\ U_\blacktriangleright^\mathcal{F} \\ Y_\blacktriangleright^\mathcal{F} \end{bmatrix} \in \mathbb{R}^{(m+p)T_\mathcal{F} \times N}. \tag{10}$$

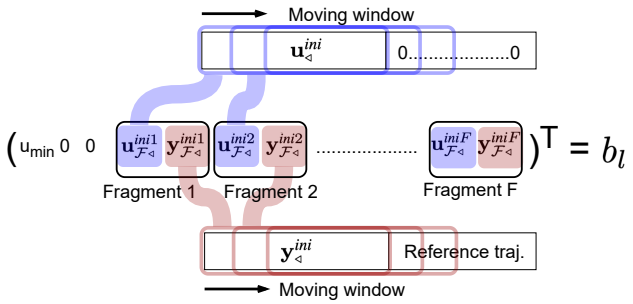

Fig. 1. Constraint-vector construction via a moving window over the extended measured-and-predicted signal. The first fragment uses the measured history; subsequent fragments obtain their past windows from the predicted trajectory through the selection matrices in (14).

Each coefficient vector $\mathbf{g}_i \in \mathbb{R}^N$ therefore represents one admissible fragment trajectory in the behavioral span of the data.

For a stacked signal of dimension d and length T, define the binary selection matrix

$$\Pi_{a,L}^{d,T} := \left[0_{L\times a}\ I_L\ 0_{L\times(T-a-L)}\right] \otimes I_d, \tag{11}$$

which extracts L consecutive samples starting at the zero-based index a. Let

$$T_e := T_{\triangleleft} + T_{\blacktriangleright}$$

be the length of the extended past-plus-future signal, and define

$$J_0^d := \begin{bmatrix} I_{dT_{\triangleleft}} \\ 0_{dT_{\blacktriangleright}\times dT_{\triangleleft}} \end{bmatrix}, \qquad J_+^d := \begin{bmatrix} 0_{dT_{\triangleleft}\times dT_{\blacktriangleright}} \\ I_{dT_{\blacktriangleright}} \end{bmatrix}. \tag{12}$$

Given the measured histories $(\mathbf{u}_{\triangleleft}^{ini}, \mathbf{y}_{\triangleleft}^{ini})$ and the predicted decision sequences $(\mathbf{u}_{\blacktriangleright}, \mathbf{y}_{\blacktriangleright})$, define the extended input and output sequences

$$\bar{u} = J_0^m \mathbf{u}_{\triangleleft}^{ini} + J_+^m \mathbf{u}_{\blacktriangleright}, \qquad \bar{y} = J_0^p \mathbf{y}_{\triangleleft}^{ini} + J_+^p \mathbf{y}_{\blacktriangleright}. \tag{13}$$

The reference trajectory is not inserted into these constraints; it appears only in the objective. This keeps $\mathbf{y}_{\blacktriangleright}$ an absolute predicted output rather than an output-error variable.

For fragment $i \in \{1,\dots,F\}$, set

$$\tau_i := (i-1)T_{\mathcal{F}_{\blacktriangleright}}.$$

The past-window and future-window selection matrices are

$$M_i^d := \Pi_{\tau_i,T_{\triangleleft}}^{d,T_e}, \qquad N_i^d := \Pi_{T_{\triangleleft}+\tau_i,T_{\mathcal{F}_{\blacktriangleright}}}^{d,T_e}. \tag{14}$$

Thus

$$u_{\triangleleft,i} = M_i^m \bar{u}, \quad y_{\triangleleft,i} = M_i^p \bar{y}, \quad u_{\blacktriangleright,i} = N_i^m \bar{u}, \quad y_{\blacktriangleright,i} = N_i^p \bar{y}. \tag{15}$$

These equations implement the moving-window idea shown in Fig. 1: the first fragment uses the measured history, whereas later fragments use the relevant suffix of the measured history and/or the predicted trajectory. The same notation also explains the heterogeneous case: one replaces τ_i by the cumulative sum $\tau_i = \sum_{j=1}^{i-1} T_{\mathcal{F}_{\blacktriangleright},j}$ and replaces N_i^d by a selector of length $T_{\mathcal{F}_{\blacktriangleright},i}$.

3.2 Decision vector and fragment equations

The F-DeePC decision vector is

$$x = \mathrm{col}(\mathbf{g}_1,\dots,\mathbf{g}_F,\mathbf{u}_{\blacktriangleright},\mathbf{y}_{\blacktriangleright},\boldsymbol{\varepsilon}_{\triangleleft},\boldsymbol{\varepsilon}_{\blacktriangleright},\boldsymbol{\varepsilon}^{\mathrm{link}}), \tag{16}$$

where $\boldsymbol{\varepsilon}_{\triangleleft} \in \mathbb{R}_{\geqslant 0}^{pT_{\triangleleft}}$, $\boldsymbol{\varepsilon}_{\blacktriangleright} \in \mathbb{R}_{\geqslant 0}^{pT_{\blacktriangleright}}$, and $\boldsymbol{\varepsilon}^{\mathrm{link}} \in \mathbb{R}_{\geqslant 0}^{pT_{\triangleleft}(F-1)}$. The slack $\boldsymbol{\varepsilon}_{\triangleleft}$ relaxes the measured initial-output consistency of the first fragment. The slack $\boldsymbol{\varepsilon}_{\blacktriangleright}$ relaxes the future-output equation of all fragments. The slack $\boldsymbol{\varepsilon}^{\mathrm{link}}$ relaxes only the output part of the inter-fragment links for fragments $2,\dots,F$. All input consistency constraints are kept hard. The number of decision variables, before any epigraph variables for $\|\mathbf{g}\|_1$, is

$$n_x = FN + (m+2p)T_{\blacktriangleright} + FpT_{\triangleleft}. \tag{17}$$

For each fragment, split the affine right-hand side of the behavioral equation into a part depending on the decision variables and one depending on the measured history:

$$B_i^u := \begin{bmatrix} M_i^m J_+^m \\ 0_{pT_{\triangleleft}\times mT_{\blacktriangleright}} \\ N_i^m J_+^m \\ 0_{pT_{\mathcal{F}_{\blacktriangleright}}\times mT_{\blacktriangleright}} \end{bmatrix}, \quad B_i^y := \begin{bmatrix} 0_{mT_{\triangleleft}\times pT_{\blacktriangleright}} \\ M_i^p J_+^p \\ 0_{mT_{\mathcal{F}_{\blacktriangleright}}\times pT_{\blacktriangleright}} \\ N_i^p J_+^p \end{bmatrix}, \qquad d_i := \begin{bmatrix} M_i^m J_0^m \mathbf{u}_{\triangleleft}^{ini} \\ M_i^p J_0^p \mathbf{y}_{\triangleleft}^{ini} \\ 0_{mT_{\mathcal{F}_{\blacktriangleright}}} \\ 0_{pT_{\mathcal{F}_{\blacktriangleright}}} \end{bmatrix}. \tag{18}$$

Then the unrelaxed fragment equation is

$$H_{\mathcal{F}}\mathbf{g}_i = B_i^u \mathbf{u}_{\blacktriangleright} + B_i^y \mathbf{y}_{\blacktriangleright} + d_i. \tag{19}$$

Stacking all fragments gives

$$(I_F \otimes H_{\mathcal{F}})\mathbf{g} - B_u \mathbf{u}_{\blacktriangleright} - B_y \mathbf{y}_{\blacktriangleright} = d, \tag{20}$$

where $\mathbf{g} := \mathrm{col}(\mathbf{g}_1,\dots,\mathbf{g}_F)$, $B_u := \mathrm{col}(B_1^u,\dots,B_F^u)$, $B_y := \mathrm{col}(B_1^y,\dots,B_F^y)$, and $d := \mathrm{col}(d_1,\dots,d_F)$.

3.3 Slack embeddings and QP matrix

Let $q_F := (m+p)T_{\mathcal{F}}$ be the number of rows in one fragment equation. Define the row selectors for the past-output and future-output rows of one fragment equation:

$$C_{\triangleleft} := [0_{pT_{\triangleleft}\times mT_{\triangleleft}}\ I_{pT_{\triangleleft}}\ 0_{pT_{\triangleleft}\times mT_{\mathcal{F}_{\blacktriangleright}}}\ 0_{pT_{\triangleleft}\times pT_{\mathcal{F}_{\blacktriangleright}}}], \tag{21}$$

$$C_{\blacktriangleright} := [0_{pT_{\mathcal{F}_{\blacktriangleright}}\times mT_{\triangleleft}}\ 0_{pT_{\mathcal{F}_{\blacktriangleright}}\times pT_{\triangleleft}}\ 0_{pT_{\mathcal{F}_{\blacktriangleright}}\times mT_{\mathcal{F}_{\blacktriangleright}}}\ I_{pT_{\mathcal{F}_{\blacktriangleright}}}]. \tag{22}$$

The corresponding slack-embedding matrices are

$$S_{\triangleleft} := e_1^F \otimes C_{\triangleleft}^{\top}, \qquad S_{\blacktriangleright} := I_F \otimes C_{\blacktriangleright}^{\top}, \tag{23}$$

$$S_{\mathrm{link}} := \begin{bmatrix} 0_{q_F\times pT_{\triangleleft}(F-1)} \\ I_{F-1} \otimes C_{\triangleleft}^{\top} \end{bmatrix}, \tag{24}$$

where e_1^F is the first canonical vector in $\mathbb{R}^F$. These matrices embed, respectively, the first measured output-window slack, the future-output slacks of all fragments, and the output-link slacks of fragments $2,\dots,F$ into the stacked fragment equations.

With the column ordering of (16), the QP constraint matrix is assembled as

$$A = \left[\begin{array}{c|c|c|c|c|c} 0 & I_{mT_{\blacktriangleright}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{pT_{\triangleleft}} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{pT_{\blacktriangleright}} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{pT_{\triangleleft}(F-1)} \\ \hline I_F\otimes H_{\mathcal{F}} & -B_u & -B_y & -S_{\triangleleft} & -S_{\blacktriangleright} & -S_{\mathrm{link}} \\ I_F\otimes H_{\mathcal{F}} & -B_u & -B_y & S_{\triangleleft} & S_{\blacktriangleright} & S_{\mathrm{link}} \end{array}\right], \tag{25}$$

where the block columns correspond to $(\mathbf{g}_1,\dots,\mathbf{g}_F)$, $\mathbf{u}_{\blacktriangleright}$, $\mathbf{y}_{\blacktriangleright}$, $\boldsymbol{\varepsilon}_{\triangleleft}$, $\boldsymbol{\varepsilon}_{\blacktriangleright}$, and $\boldsymbol{\varepsilon}^{\mathrm{link}}$. The first four block rows impose, respectively, input bounds and nonnegativity of $\boldsymbol{\varepsilon}_{\triangleleft}$, $\boldsymbol{\varepsilon}_{\blacktriangleright}$, and $\boldsymbol{\varepsilon}^{\mathrm{link}}$. The last two block rows encode the relaxed fragment equations with opposite slack signs.

Equivalently, the last two block rows impose

$$d - S_\epsilon \leqslant (I_F \otimes H_{\mathcal{F}})\mathbf{g} - B_u \mathbf{u}_{\blacktriangleright} - B_y \mathbf{y}_{\blacktriangleright} \leqslant d + S_\epsilon, \quad (26)$$

where $S_\epsilon := S_{\blacktriangleleft}\boldsymbol{\varepsilon}_{\blacktriangleleft} + S_{\blacktriangleright}\boldsymbol{\varepsilon}_{\blacktriangleright} + S_{\text{link}}\boldsymbol{\varepsilon}^{\text{link}}$. The slack vector S_ϵ has nonzero entries only in the selected output rows. Thus, output consistency is relaxed, whereas input consistency remains hard.

The corresponding bounds are

$$b_l = \text{col}(b_l^0, -\infty_{Fq_F}, d), \qquad b_u = \text{col}(b_u^0, d, +\infty_{Fq_F}), \quad (27)$$

where

$$b_l^0 = \text{col}(\mathbf{1}_{T_{\blacktriangleright}} \otimes \mathbf{u}_{\min}, 0_{pT_{\blacktriangleleft}}, 0_{pT_{\blacktriangleright}}, 0_{pT_{\blacktriangleleft}(F-1)}), \quad (28)$$

$$b_u^0 = \text{col}(\mathbf{1}_{T_{\blacktriangleright}} \otimes \mathbf{u}_{\max}, +\infty_{pT_{\blacktriangleleft}}, +\infty_{pT_{\blacktriangleright}}, +\infty_{pT_{\blacktriangleleft}(F-1)}). \quad (29)$$

If hard or softened output box constraints are required, they can be appended as additional rows involving $\mathbf{y}_{\blacktriangleright}$ and, if softened, additional output-constraint slacks.

4. EXPERIMENTAL EVALUATION

We compare F-DeePC with DeePC and S-DeePC on two Single-Input Single-Output (SISO) benchmarks: a noisy two-mass-spring-damper LTI system from (O'Dwyer et al., 2023) and a damped double pendulum.

Performance is measured by Sum of Set-Point Tracking Errors (SPTE), the accumulated absolute tracking error, as in (O'Dwyer et al., 2023). Each experiment is repeated 100 times with repetition-indexed random seeds. The DeePC and S-DeePC baselines use the companion code of (O'Dwyer et al., 2023); F-DeePC is implemented in Python using NumPy (Harris et al., 2020) and OSQP (Stellato et al., 2020). The implementation is available online [1] and archived on Zenodo [2].

4.1 Hyperparameter analysis

F-DeePC introduces the homogeneous fragment prediction horizon $T_{\mathcal{F}_{\blacktriangleright}}$ in addition to the common prediction horizon, initial horizon, and regularization term λ_g. We tune $T_{\mathcal{F}_{\blacktriangleright}}$ and λ_g on the noiseless LTI benchmark over 30 repetitions. As shown in Fig. 2, the best tracking performance is obtained with $T_{\mathcal{F}_{\blacktriangleright}} = 3$, consistent with the system controllability (Mishra et al., 2021), while λ_g has little effect. We therefore set $\lambda_g = 6.5$ and $T_{\mathcal{F}_{\blacktriangleright}} = 3$.

4.2 Use case 1: LTI system with noise

The first benchmark is the noisy two-mass-spring-damper LTI system shown in Fig. 3. It consists of two masses $m_1 = 0.5$kg and $m_2 = 1.5$kg connected by springs $k_1 = k_2 = 2$N/m and dampers $c_1 = c_2 = 1$N s/m. The input force acts on the first mass, and the output is the position of the second mass. Its dynamics are

$$\begin{aligned} \dot{x} &= Ax + B(u + n_{\text{rand}} + n_{\sin}), \\ y &= Cx + n_{\text{meas}}, \end{aligned} \quad (30)$$

where $u \in [-1, 1]$N, $n_{\text{rand}} \sim \mathcal{U}(-0.15, 0.15)$N, $n_{\sin} = 0.2\sin(2\pi \cdot 0.01t)$N $+ 0.2$N, and $n_{\text{meas}} \sim \mathcal{N}(0, 0.1)$m.

Dataset generation. The dataset is generated once using a discretized LTI system with $\Delta t = 1$s, zero initial conditions, and a persistently exciting input (Willems et al., 2005) in $[-1, 1]$N, randomly updated every 10 time steps. After collecting L samples, the data are arranged into a Hankel matrix as in (2).

[1] https://github.com/Crylab/DeepC-Fragmented
[2] https://doi.org/10.5281/zenodo.15065011

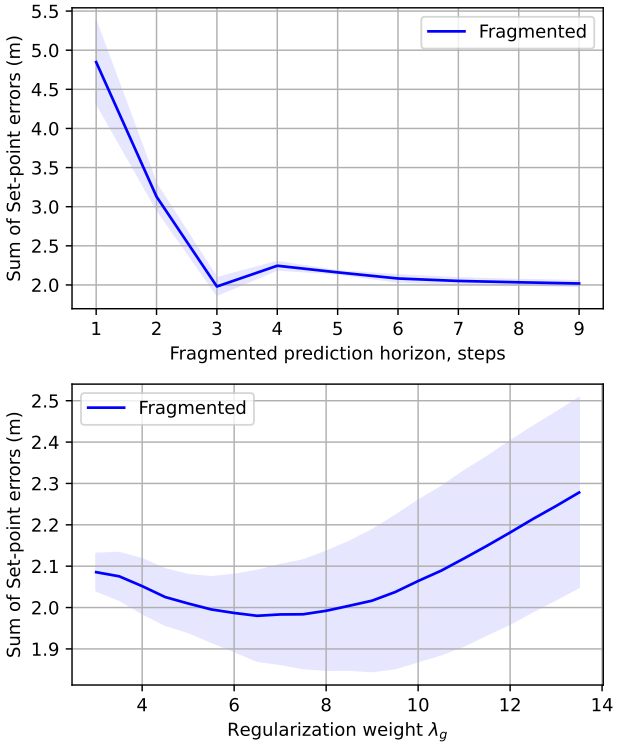

Fig. 2. Effect of $T_{\mathcal{F}_{\blacktriangleright}}$ and λ_g on the noiseless LTI system. Shaded areas show ± 1 standard deviation over 30 repetitions.

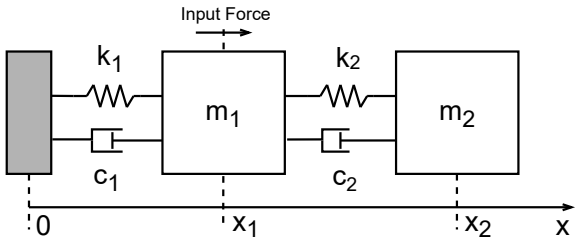

Fig. 3. Two-mass-spring-damper SISO benchmark.

Performance evaluation. Results are summarized in Fig. 4, with the case $L = 50$ detailed in Fig. 5. Performance improves with dataset size. S-DeePC shows a stronger error-reduction rate, consistent with the advantage of indirect predictors at high noise-to-signal ratios (Dörfler et al., 2023; Krishnan and Pasqualetti, 2021). However, F-DeePC performs best with smaller datasets: DeePC becomes unstable for $L = 40$ and S-DeePC struggles for $L = 30$, while F-DeePC remains stable. For $L = 50$, F-DeePC and S-DeePC perform similarly and both outperform DeePC.

4.3 Use case 2: Nonlinear double pendulum

The second benchmark is a double pendulum with damping at the interconnecting joint, shown in Fig. 6. We evaluate damping coefficients $\zeta \in \{0.01, 0.1, 1.0\}$N m s. Larger

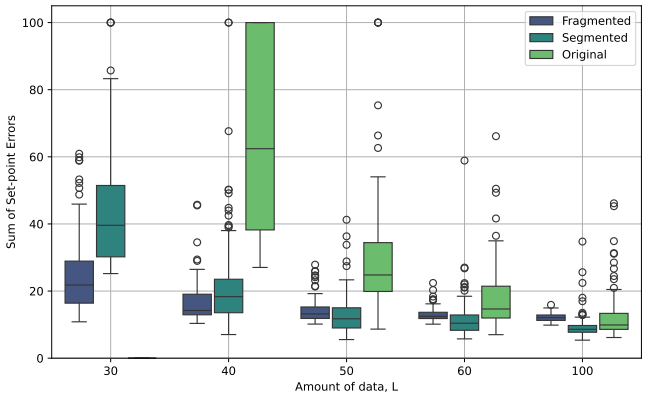

Fig. 4. Boxplots of SPTE for the noisy LTI system with varying dataset sizes.

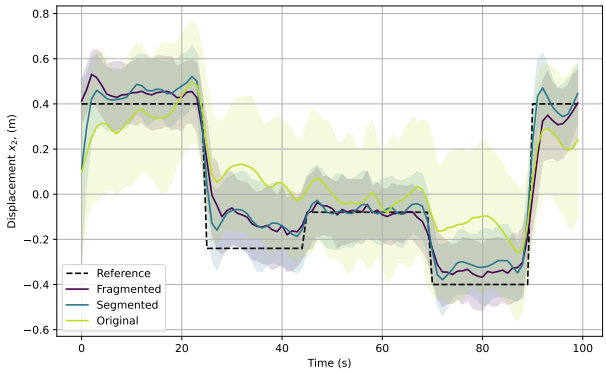

Fig. 5. Tracking performance with $L = 50$. Solid lines show means over 100 runs, shaded areas show mean $\pm 1\sigma$, and the dashed line is the reference.

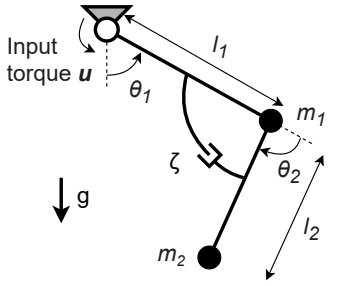

Fig. 6. Damped double-pendulum benchmark.

ζ values make the dynamics closer to a single pendulum, while smaller values yield more complex behavior, as shown in Fig. 7.

Dataset generation. For the nonlinear benchmark, trajectories are collected using a stabilizing auxiliary PD controller with gravity compensation, following related data-generation strategies (Alsalti et al., 2023; Persson et al., 2026):

$$u_{PD}(\theta_1, \dot{\theta}_1, \theta_2) = l_1 m_1 g \sin(\mathbf{r}) + m_2 g(l_1 \sin(\mathbf{r}) + l_2 \sin(\theta_2)) + 20(\mathbf{r} - \theta_1) - 10\dot{\theta}_1. \tag{31}$$

Random set-points, each lasting 10 steps, are used as references. The auxiliary controller is used only for data generation. The system is simulated for L steps with $\Delta t = 0.2$s, and the resulting data are arranged into a Hankel matrix as in (2).

Performance evaluation. Fig. 8 shows that lower damping makes the task harder for all methods, while F-DeePC consistently achieves lower median error and variance, especially in the most nonlinear cases. Fig. 9 shows that increasing the dataset size benefits F-DeePC the most. In contrast, S-DeePC performs worse because its least-squares predictor tends to treat nonlinear effects as noise. Overall, F-DeePC gives the best results across damping values and dataset sizes.

5. CONCLUSION

This work introduced F-DeePC, a method that partitions the prediction horizon into smaller fragments based on a direct formulation of DeePC. By explicitly constructing the constraint matrix, F-DeePC reduces the dataset requirements and median error while allowing independent adjustment of the horizon lengths in each fragment.

Experiments on a noisy LTI system and a damped double pendulum show that F-DeePC consistently outperforms state-of-the-art methods for small datasets. Unlike the indirect formulation of S-DeePC, F-DeePC adopts a direct formulation, trading higher sensitivity to large noise-to-signal ratios in LTIs (Dörfler et al., 2023) for better performance for complex nonlinear systems (Krishnan and Pasqualetti, 2021) across all dataset sizes.

In conclusion, F-DeePC is a promising approach for applying data-driven control to systems with limited data. Future research will focus on MIMO systems and practical applications in autonomous vehicles and quadrotors.

REFERENCES

Abolpour, R., Khayatian, A., and Dehghani, M. (2024). Simultaneous model prediction and data-driven control with relaxed assumption on the model. *ISA Trans.*, 145.

Alpago, D., Dörfler, F., and Lygeros, J. (2020). An extended kalman filter for data-enabled predictive control. *IEEE Control. Syst. Lett.*, 4(4).

Alsalti, M. et al. (2023). Data-based control of feedback linearizable systems. *IEEE Trans. Autom. Control*.

Alsalti, M. et al. (2024). Sample- and computationally efficient data-driven predictive control. In *Eur. Contr. Conf. (ECC)*.

Berberich, J., Köhler, J., Müller, M.A., and Allgöwer, F. (2021). Data-driven model predictive control with stability and robustness guarantees. *IEEE Trans. Autom. Control*, 66(4).

Bouffard, P., Aswani, A., and Tomlin, C. (2012). Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results. In *IEEE Int. Conf. Robotics and Autom. (ICRA)*.

Chiuso, A. et al. (2025). Harnessing uncertainty for a separation principle in direct data-driven predictive control. *Autom.*, 173.

Coulson, J., Lygeros, J., and Dörfler, F. (2019). Data-enabled predictive control: In the shallows of the DeePC. In *Eur. Contr. Conf. (ECC)*.

Dörfler, F., Coulson, J., and Markovsky, I. (2023). Bridging direct and indirect data-driven control formulations

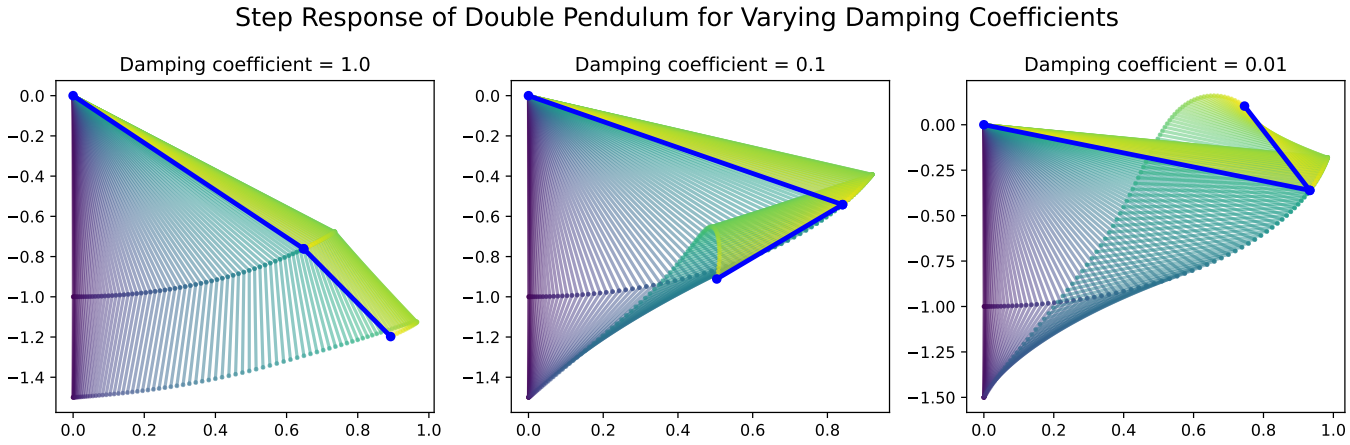

Fig. 7. Step response of the double pendulum for different damping coefficients: $\zeta = 1\mathrm{N\,m\,s}$, $0.1\mathrm{N\,m\,s}$, and $0.01\mathrm{N\,m\,s}$. Lower damping yields more complex, eventually chaotic, dynamics.

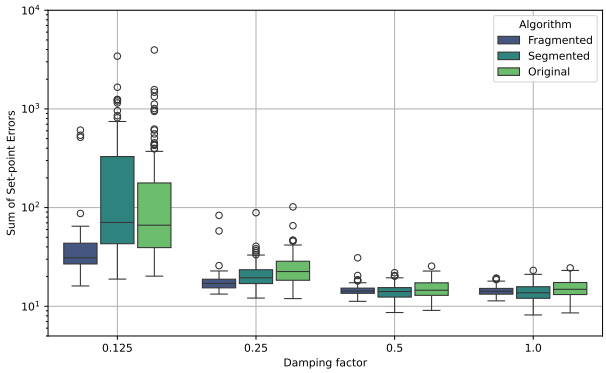

Fig. 8. Boxplots of SPTE for the double pendulum with $L = 100$ and different damping values ζ.

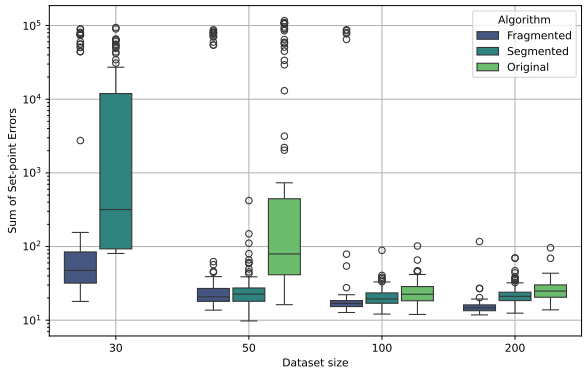

Fig. 9. Boxplots of SPTE with $\zeta = 0.25\mathrm{N\,m\,s}$ for different dataset sizes.

via regularizations and relaxations. *IEEE Trans. Autom. Control*, 68(2).

Dörfler, F., Tesi, P., and Persis, C.D. (2022). On the role of regularization in direct data-driven LQR control. In *IEEE Conf. Dec. and Contr. (CDC)*.

Elokda, E. et al. (2021). Data-enabled predictive control for quadcopters. *Int. J. Rob. and Nonlin. Contr.*, 31(18).

Harris, C.R. et al. (2020). Array programming with numpy. *Nature*, 585(7825).

Huang, L., Coulson, J., Lygeros, J., and Dörfler, F. (2022). Decentralized data-enabled predictive control for power system oscillation damping. *IEEE Trans. Contr. Syst. Tech.*, 30(3).

Huang, L. et al. (2024). Robust and kernelized data-enabled predictive control for nonlinear systems. *IEEE Trans. Contr. Syst. Tech.*, 32(2).

Krishnan, V. and Pasqualetti, F. (2021). On direct vs indirect data-driven predictive control. In *IEEE Conf. Dec. and Contr. (CDC)*.

Markovsky, I. and Rapisarda, P. (2008). Data-driven simulation and control. *Int. J. Control*, 81(12).

Markovsky, I. et al. (2023). Data-driven control based on the behavioral approach: From theory to applications in power systems. *IEEE Control Systems*, 43(5).

Mishra, V.K., Markovsky, I., and Grossmann, B. (2021). Data-driven tests for controllability. *IEEE Control. Syst. Lett.*, 5(2).

O'Dwyer, E. et al. (2023). Data-driven predictive control with improved performance using segmented trajectories. *IEEE Trans. Contr. Syst. Tech.*, 31(3).

Persson, N. et al. (2026). An adaptive data-enabled policy optimization approach for autonomous bicycle control. *IEEE Trans. Contr. Syst. Tech.*

Pillonetto, G. et al. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Autom.*, 50(3).

Rapisarda, P. and Pal, D. (2023). Informativity for identification for 2D state-representable autonomous systems, with applications to data-driven simulation. In *IEEE Conf. Dec. and Contr. (CDC)*.

Seuret, A., Albea, C., and Gordillo, F. (2023). Linear matrix inequality relaxations and its application to data-driven control design for switched affine systems. *Int. J. Rob. and Nonlin. Contr.*, 33(12).

Shafieezadeh-Abadeh, S. et al. (2019). Regularization via mass transportation. *J. Mach. Learn. Res.*, 20.

Stellato, B. et al. (2020). OSQP: an operator splitting solver for quadratic programs. *Math. Program. Comput.*

van Waarde, H.J. et al. (2020). Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control. Syst. Lett.*, 4(3).

Willems, J.C. et al. (2005). A note on persistency of excitation. *Syst. Control. Lett.*, 54(4).