



# A computationally efficient reformulation for Data-Enabled Predictive Control

Alexandre Faye-Bedrin, Stanislav Aranovskiy, Paul Chauchat, Romain  
Bourdais

## ► To cite this version:

Alexandre Faye-Bedrin, Stanislav Aranovskiy, Paul Chauchat, Romain Bourdais. A computationally efficient reformulation for Data-Enabled Predictive Control. 2024. hal-04542440

**HAL Id: hal-04542440**

**<https://hal.science/hal-04542440>**

Preprint submitted on 11 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A computationally efficient reformulation for Data-Enabled Predictive Control

Alexandre Faye-Bédin<sup>1</sup>, Stanislav Aranovski<sup>1</sup>, Paul Chauchat<sup>2</sup>, and Romain Bourdais<sup>1</sup>

**Abstract**—This work investigates the computational efficiency of Data-Enabled Predictive Control (DeePC) reformulations. Based on Willems’ fundamental lemma, this control method uses Hankel matrices to represent system dynamics. The size—in particular the number of columns—of these Hankel matrices can incur a significant computational complexity, which has seen several attempts at being reduced. We propose a reformulation of DeePC aiming for lower complexity and show how it allows for exponential forgetting of online collected data. The method’s effectiveness is illustrated by results obtained both in simulation and experimentally.

## I. INTRODUCTION

Willems fundamental lemma [1] is a powerful result for linear time-invariant (LTI) systems analysis and control. It states that every trajectory of a LTI system can be derived from a Hankel matrix, built from a longer trajectory under a few conditions, such as the persistence of excitation of the input. This implicit system representation can have many uses [2], in particular, simulation [3] and control [4]. In recent years, this approach sparked a lot of interest and led to the development of analysis and control methods [5], [6]. Notably, data-enabled predictive control (*DeePC*) methods have been developed [7], [8]. Different flavors include offset-free tracking [9], [10], online data update [11], multiple shooting [12], and distributional robustness [13], [14]. Some of these schemes use a bi-level formulation of the Optimal Control Problem (OCP) to decouple “model” fitness and desired performance [15], [16]. The general method has been successfully applied to different systems, such as quadcopters [17], power converters [18], [19], or building temperature regulation [16].

In (predictive) control applications, computational complexity is a fundamental question: it is necessary to solve an OCP fast enough, e.g., at each sample time, in order to be able to apply it. Within the DeePC framework, OCP can have very high complexity, as they compute the best linear combination of numerous trajectories. In other words, the quantity of data has a direct (negative) influence on the computational cost. Because of this, several attempts have been made to reduce this cost: among them, using a singular value decomposition (SVD) of the mosaic-Hankel matrix to reduce the dimension of the optimal variable seems the most recurring [20], [10]. Some authors use (inverse) Fourier transforms to perform fast Hankel matrix-vector multiplication [11], [21], although it seems restricted to

specific optimization methods. In [22], the authors propose an optimization method tailored to a specific OCP. SPC (Subspace Predictive Control [23]) could be cited as a closely related (and computationally cheaper) method that in some cases, is equivalent to DeePC [24], [25]; however, it is not as straightforward to perform online updates. As a last example, GPC (Generalized Predictive Control) is a compromise between SPC and DeePC [26], in terms of both flexibility and compute cost.

We introduce a new reformulation for various DeePC OCP that does not rely on SVD, nor any specific optimization method. This reformulation is equivalent (in terms of optimal value function and optimal input) to a reference OCP, and we demonstrate it gives the same results while taking less time to compute. The proposed reformulation also opens new possibilities regarding data managing, as it allows efficient incremental updates of the underlying matrices, and thus the possibility to apply exponential forgetting of data. Experiments illustrate the advantages of the proposed method.

The rest of this paper is organized as follows. Section II presents the Hankel-based DeePC problem formulation. In Section III, we show our main contribution, that is an equivalent reformulation with reduced dimension. We also show how the underlying matrices can be iteratively constructed and how exponential forgetting can be easily implemented. Experimental validation of the proposed method and comparison to the standard one is given in Section IV. A conclusion is given in Section V.

## II. PRELIMINARIES

### Notation

- The operator  $\text{col}(\cdot)$  stacks up its vector arguments.
- For a sequence  $\{z_k\}_{k=m}^M$  and  $a, b \in [m, M]$  we use the following notation for a stacked window:

$$z_{[a,b]} = \text{col}(z_a, z_{a+1}, \dots, z_b)$$

With a slight abuse of notations, we write  $z$  for  $z_{[m,M]}$ .

- For a sequence  $\{z_k\}_{k=1}^N$  with  $z_k \in \mathbb{R}^{n_z}$ , we denote the Hankel matrix  $H_L(z) \in \mathbb{R}^{L n_z \times (N-L+1)}$  as

$$H_L(z) = \begin{bmatrix} z_{[1,L]} & z_{[2,L+1]} & \dots & z_{[N-L+1,N]} \end{bmatrix}.$$

- A vector or sequence computed at time  $t$  is denoted by  $\cdot(t)$ , e.g.,  $\{z_k(t)\}_{k=m}^M$ .
- A vector of ones of length  $q$  is denoted by  $\mathbb{1}_q$ .
- The Kronecker product is denoted by  $\otimes$ .
- Quadratic forms are denoted:  $\|x\|_K^2 = x^\top K x, K \succeq 0$

<sup>1</sup> IETR, CentraleSupélec, Rennes, France  
surname.name@centralesupelec.fr

<sup>2</sup> Aix-Marseille Univ, CNRS, LIS, Marseille, France  
paul.chauchat@lis-lab.fr

- For a (semi-)positive definite matrix  $K$ ,  $K^{\frac{1}{2}}$  is the only (semi-)positive definite matrix such that  $K = (K^{\frac{1}{2}})^2$ .
- The Moore-Penrose inverse of a matrix  $A$  is denoted  $A^\dagger$ . It satisfies

$$A^\dagger A A^\top = A^\top. \quad (1)$$

- The range space of a matrix  $A$  is denoted  $\text{im}(A)$ , and its null space is  $\ker(A)$ .

#### A. Willems' Fundamental Lemma

We recall Willems' fundamental lemma for affine systems [27], [28], as we use it in examples. However, the "original" version for LTI systems [1] would also fit in the reformulations.

Let  $\mathcal{B}$  a controllable and observable affine system of order  $n$ , with  $n_u$  inputs and  $n_y$  outputs.

*Definition 1:* A sequence  $\{z_k\}_{k=1}^N$  with  $z_k \in \mathbb{R}^q$  is said to be persistently exciting of order  $M$  if

$$\text{rank}(H_M(z)) = qM.$$

Let  $(u^d = \{u_k^d\}_{k=1}^N, y^d = \{y_k^d\}_{k=1}^N)$  be a series of inputs and outputs of  $\mathcal{B}$ .

*Theorem 1:* If  $u^d$  is persistently exciting of order  $L + n + 1$ , then  $(u, y)$  with  $u = \text{col}(u_1, \dots, u_L)$  and  $y = \text{col}(y_1, \dots, y_L)$  is a trajectory of length  $L$  of  $\mathcal{B}$  if and only if

$$\exists \alpha \in \mathbb{R}^{N-L+1}, \begin{bmatrix} \mathbb{1}_{N-L+1}^\top \\ H_L(u^d) \\ H_L(y^d) \end{bmatrix} \alpha = \begin{bmatrix} 1 \\ u \\ y \end{bmatrix}. \quad (2)$$

#### B. Data Enabled Predictive Control

Consider the following formulation that can be used for some of DeePC problems, e.g., [8], [14], [16], [18]:

$$\text{Compute } f(Wg^*) \text{ with} \quad (3a)$$

$$g^* \in \arg \min_g J(Wg) + h(\|g\|_K) \quad (3b)$$

$$\text{st. } Wg \in \mathcal{W} \quad (3c)$$

with  $h$  increasing and  $K \succ 0$ .

In the typical setting,  $W$  contains a (mosaic) Hankel matrix,  $J$  represents a tracking objective, and  $h(\|g\|_K)$  is a regularization term;  $f$  retrieves the optimal input  $u^*$  from  $Wg^*$ .

For example, consider the following illustrative DeePC tracking problem based on Theorem 1:

Compute  $\bar{u}$  with

$$g^* \in \arg \min_g \sum_{k=0}^{L-1} \|\bar{y}_k - y^r\|_Q^2 + \lambda_\sigma \|\sigma\|_2^2 + \lambda_g \|g\|_2^2 \quad (4a)$$

$$\text{st. } \begin{cases} u_{\min} \leq \bar{u}_{[0,L]} \leq u_{\max} \\ \bar{u}_{[-n,-1]} = u_{[t-n,t-1]} \end{cases} \quad (4b)$$

$$\begin{cases} \begin{bmatrix} \mathbb{1}_{N-L-n+1}^\top \\ H_{L+n}(u^d) \\ H_{L+n}(y^d) \end{bmatrix} g = \begin{bmatrix} 1 \\ \bar{u} \\ \bar{y} \end{bmatrix} \\ \bar{y}_{[-n,-1]} = y_{[t-n,t-1]} + \sigma \end{cases} \quad (4c)$$

with  $Q, \lambda_\sigma, \lambda_g$  cost parameters, and  $u_{\min}, u_{\max}$  input bounds. In this problem,  $y^r$  is the reference output,  $\bar{u}$  and  $\bar{y}$  are the predicted (planned) trajectory, and  $\sigma$  is a slack variable allowing for small deviation from the past measurements; these variables are derived from  $g$  in (4c).

Then the DeePC problem (4) can be written as (3) choosing

$$W = \begin{bmatrix} \mathbb{1}_{N-L-n+1}^\top \\ H_{L+n}(u^d) \\ H_{L+n}(y^d) \end{bmatrix}; \quad h(\|g\|_K) = \lambda_g \|g\|_2^2 \quad (5)$$

$$J(Wg) = \|H_L(y^d)g - \bar{y}^r\|_Q^2 = \|S_y Wg - \bar{y}^r\|_Q^2 \quad (6)$$

$$f(Wg) = H_L(u^d)g = S_u Wg \quad (7)$$

with

$$\bar{y}^r = \begin{bmatrix} y_{[t-n,t-1]} \\ \mathbb{1}_L \otimes y^r \end{bmatrix}; \quad \bar{Q} = \begin{bmatrix} \lambda_\sigma \mathbb{I}_{nn_y} & 0 \\ 0 & \mathbb{I}_L \otimes Q \end{bmatrix}$$

$$S_u = \begin{bmatrix} 0 & \mathbb{I}_{Ln_u} & 0 \end{bmatrix}; \quad S_y = \begin{bmatrix} 0 & 0 & \mathbb{I}_{Ln_y} \end{bmatrix}.$$

Notice that  $W$  not only contains the data  $(u^d, y^d)$  but also constraints that  $g$  must satisfy. Rows of  $W$  can then be selected to retrieve Hankel matrices, in our case with  $S_u, S_y$ .

### III. REDUCING THE PROBLEM SIZE

The application of DeePC relies on solving quadratic problems involving matrix  $W$  from (5). Let  $\#rows(W) \times \#cols(W)$  be the dimension of  $W$ .  $\#cols(W)$  depends on  $N$ , the size of the dataset  $(u^d, y^d)$ , which is in general much larger than the other dimensions, such that  $\#cols(W) \gg \#rows(W)$ . Therefore, the larger the dataset, the larger the computational burden. This is a strong restriction to the applicability of DeePC, since more data usually means a better representation of the system [14], [29]. In this paper, we propose a reformulation of Problem 3 which relies on a matrix of dimension  $\#rows(W) \times \#rows(W)$ .

#### A. Range Space Equivalence

We heavily rely on the following lemma to derive our problem reformulation:

*Lemma 1 (Range space equivalence):* Let  $K \in \mathbb{R}^{n \times n}$  a positive definite matrix. Then, for any matrix  $A \in \mathbb{R}^{m \times n}$ :

$$\text{im}(A) = \text{im}(AKA^\top). \quad (8)$$

In particular, with  $K = \mathbb{I}$ :

$$\text{im}(A) = \text{im}(AA^\top). \quad (9)$$

*Proof:*

Since  $K \succ 0$ , for any  $x \in \mathbb{R}^m$  it holds

$$AKA^\top x = 0 \implies \|A^\top x\|_K = 0 \implies A^\top x = 0$$

yielding

$$A^\top x = 0 \iff AKA^\top x = 0,$$

and thus  $\ker(A^\top) = \ker(AKA^\top)$ . Using the fact that  $\text{im}(A) = (\ker(A^\top))^\perp$ , we obtain

$$\text{im}(A) = (\ker(A^\top))^\perp = (\ker(AKA^\top))^\perp = \text{im}(AKA^\top). \quad \blacksquare$$

### B. Problem reformulation

We propose the following reformulation of Problem 3, which can reduce the problem size, through the following theorem:

*Theorem 2 (Problem equivalence):* Problem 3 is equivalent to Problem 10, i.e. they have the same output and optimal function value:

$$\text{Compute } f(G\alpha^*) \text{ with} \quad (10a)$$

$$\alpha^* \in \arg \min_{\alpha} J(G\alpha) + h(\|\alpha\|_G) \quad (10b)$$

$$\text{st. } G\alpha \in \mathcal{W} \quad (10c)$$

with  $G = WK^{-1}W^\top$ .

*Proof:* We will show the equivalence by showing that the solutions  $g^*$  of Problem 3 and  $\alpha^*$  of Problem 10 satisfy  $Wg^* = G\alpha^*$  and  $\|g^*\|_K = \|\alpha^*\|_G$ .

Since  $K \succ 0$ , so is  $K^{-1}$ . Lemma 1 ensures

$$\forall g, \exists \alpha, Wg = WK^{-1}W^\top \alpha = G\alpha \quad (11)$$

Let  $g, \alpha$  satisfying (11), and rewrite  $\mathbb{I} = K^{-\frac{1}{2}}K^{\frac{1}{2}}$  and  $K^{-1} = K^{-\frac{1}{2}}K^{-\frac{1}{2}}$ . We then have

$$WK^{-\frac{1}{2}}K^{\frac{1}{2}}g = WK^{-\frac{1}{2}}K^{-\frac{1}{2}}W^\top \alpha \quad (12)$$

By definition of the Moore-Penrose inverse, for any solution of the given linear system  $Ax = b$ , there exists  $c$  such that  $x = A^\dagger b + (\mathbb{I} - A^\dagger A)c$ . Applying this to (12), with  $A = WK^{-\frac{1}{2}}$ , we get

$$\exists c, K^{\frac{1}{2}}g = \left(WK^{-\frac{1}{2}}\right)^\dagger WK^{-\frac{1}{2}}K^{-\frac{1}{2}}W^\top \alpha \quad (13)$$

$$+ \left(\mathbb{I} - \left(WK^{-\frac{1}{2}}\right)^\dagger WK^{-\frac{1}{2}}\right)c \quad (14)$$

$$= K^{-\frac{1}{2}}W^\top \alpha + Pc \quad (15)$$

with  $P = \left(\mathbb{I} - \left(WK^{-\frac{1}{2}}\right)^\dagger WK^{-\frac{1}{2}}\right)$ , where the last equality follows from (1). Then, fixing  $c$

$$\begin{aligned} \|g\|_K^2 &= \|K^{\frac{1}{2}}g\|_2^2 \\ &= \|K^{-\frac{1}{2}}W^\top \alpha + Pc\|_2^2 \\ &= \|K^{-\frac{1}{2}}W^\top \alpha\|_2^2 + \|Pc\|_2^2 \end{aligned}$$

since, by definition,  $PK^{-\frac{1}{2}}W^\top = WK^{-\frac{1}{2}}P = 0$ . Finally

$$\begin{aligned} \|g\|_K^2 &= \|W^\top \alpha\|_{K^{-1}}^2 + \|Pc\|_2^2 \\ &= \|\alpha\|_G^2 + \|Pc\|_2^2. \end{aligned}$$

Then, for a given value of  $Wg = G\alpha$ , the minimum of  $\|g\|_K$  (as well as the minimum of  $h(\|g\|_K)$ ) is attained if  $Pc = 0$ . We then have, for solutions  $g^*$  and  $\alpha^*$ :

$$\begin{cases} f(Wg^*) &= f(G\alpha^*) \\ J(Wg^*) + h(\|g^*\|_K) &= J(G\alpha^*) + h(\|\alpha^*\|_G). \end{cases} \quad (16)$$

### C. Iterative construction

It turns out that problem reformulation given in Theorem 2 also has an advantage in terms of memory requirements. Indeed, we show in this section that  $G$  can be constructed incrementally.

We now take  $K$  diagonal, and denote  $k_i > 0$  the  $i$ -th diagonal element of  $K$ . Typically,  $W$  would contain Hankel matrices of input/output samples:

$$W = \begin{bmatrix} A \\ H_L(u^d) \\ H_L(y^d) \end{bmatrix} \quad (17)$$

with  $A$  describing constraints, e.g. as in (2).

We aim at “updating”  $W$ , that is, to add (or remove) a column. This typically happens when a new data sample is appended to  $(u^d, y^d)$ , and the corresponding Hankel matrices gain a new column.

Notice how  $G$  is constructed, with  $w_i$  the  $i$ -th column of  $W$ :

$$\begin{aligned} G &= WK^{-1}W^\top \\ &= \sum_{i=1}^{\#cols(W)} \frac{1}{k_i} w_i w_i^\top \end{aligned}$$

This means appending a column  $w_{new}$ , associated to a weight  $k_{new}$ , to  $W$  amounts to a matrix addition with  $G$ :

$$\begin{aligned} W_{new} &= [W \quad w_{new}] \\ \implies G_{new} &= W_{new}K_{new}^{-1}W_{new}^\top \\ &= WK^{-1}W^\top + \frac{1}{k_{new}}w_{new}w_{new}^\top \end{aligned}$$

and we recognize

$$G_{new} = G + \frac{1}{k_{new}}w_{new}w_{new}^\top \quad (18)$$

A column  $w_{rem}$  (with the weight  $k_{rem}$ ) can also be removed from  $W$ : writing  $W = [W_{new} \quad w_{rem}]$  and combining with (18) yields

$$G_{new} = G - \frac{1}{k_{rem}}w_{rem}w_{rem}^\top \quad (19)$$

Note that—in general—neither  $W, K$  (nor  $K^{-1}$ ) need to be explicitly computed nor stored at any time. This allows potentially huge datasets while keeping a reasonable space and time complexity.

The process can be tweaked if  $K$  is to be modified (other than adding/removing columns), as we show next with exponential forgetting.

### D. Exponential forgetting

Using the iterative construction of matrix  $G$ , achieving what can be interpreted as exponential forgetting is possible. Exponential forgetting is widely used in different contexts, including recursive least squares, state observers, and adaptive control. Its appeal is important since it provides a simple framework for adaptive models, and it is well understood, see [30] and references therein.

Let  $0 < \rho < 1$  be the forgetting factor. We denote with  $w_{-i}$  the  $i$ -th most recent column of  $W$ : if new columns are appended to the right of  $W$ , we have  $w_{-i} = w_{\#cols(W)+1-i}$  (similarly, we define  $k_{-i} = k_{\#cols(W)+1-i}$ ).

Intuitively,  $\rho^j$  represents “how much” we should use  $w_{-j}$ . This can be translated as  $k_{-j} = \frac{k_0}{\rho^j}$ , with some  $k_0 > 0$ : older samples are more penalized. We take  $k_0 = \rho$  for simplicity, without loss of generality, since  $h$  can apply any scalar factor. Then at each time step  $t$ ,  $W$  and  $K$  are updated as such:

$$\begin{aligned} W_{t+1} &= [W_t \quad w_{t+1}] \\ K_{t+1} &= \frac{1}{\rho} \begin{bmatrix} K_t & 0 \\ 0 & \rho \end{bmatrix} \end{aligned}$$

In turn,  $G$  can be updated this way:

$$G_{t+1} = W_{t+1} K_{t+1}^{-1} W_{t+1}^\top \quad (20)$$

$$= \rho W_t K_t^{-1} W_t^\top + w_{t+1} w_{t+1}^\top \quad (21)$$

$$= \rho G_t + w_{t+1} w_{t+1}^\top. \quad (22)$$

Note that, if  $\{w_t\}_{t>0}$  is bounded, then so is  $\{G_t\}_{t>0}$ , even if no column removal is performed.

However, one concern remains: if numerous similar new entries are added, informative data is more and more compressed, and the matrix  $G$  becomes ill-conditioned, making (10) hard to solve numerically. This is a common problem of the lack of excitation and can be addressed using, e.g., variable forgetting factor [30], or adding a second data matrix [31], as is explained in Section IV-B.

#### E. Extension to mosaic-Hankel matrices

1) *Considering two datasets*: In practice, it is useful to formulate Problem 3 using two datasets (i.e., two separate trajectories instead of a long one) to help overcome the lack of excitation problem [31], [32]. In this case,  $W$  becomes a mosaic-Hankel matrix:

$$W_t = \begin{bmatrix} \mathbb{1}_{N-L-n+1}^\top & \mathbb{1}_{N_o-L-n+1}^\top \\ H_{L+n}(u^d) & H_{L+n}(u_{[t-N_o, t-1]}) \\ H_{L+n}(y^d) & H_{L+n}(y_{[t-N_o, t-1]}) \end{bmatrix} \quad (23)$$

Typically,  $(u_k^d, y_k^d)_{k \in [1, N]}$  is first initialized to satisfy Theorem 1 conditions, e.g., by injecting a probe signal, while  $(u_k, y_k)_{k \in [t-N_o, t-1]}$  is dynamically updated to adapt to possible operating points variation. In a certain sense, the first (initial) dataset can be considered as a *backup* one for the case when the updated dataset is not exciting.

The proposed reformulation directly extends to this setting. Indeed, since Theorem 2 makes no assumption on the structure of  $W$ , it also holds in this case.

2) *Iterative construction and exponential forgetting*: The iterative construction, with exponential forgetting, can also be adapted to the case of mosaic Hankel-matrices. Consider

$$W_t = \begin{bmatrix} \mathbb{1}_{N-L-n+1}^\top & \mathbb{1}_{t-L-n}^\top \\ H_{L+n}(u^d) & H_{L+n}(u_{[1, t-1]}) \\ H_{L+n}(y^d) & H_{L+n}(y_{[1, t-1]}) \end{bmatrix} \quad (24)$$

Again, the first dataset  $(u_k^d, y_k^d)_{k \in [1, N]}$  is initialized at the beginning of the experiment; the second  $(u_k, y_k)_{k \in [1, t-1]}$  is

dynamically updated. To implement the iterative construction of  $G$ , we first split  $W$  and  $K$ :

$$W_t = [W_{bak} \quad W_{exp, t}] \quad (25)$$

$$K_t = \begin{bmatrix} K_{bak} & 0 \\ 0 & K_{exp, t} \end{bmatrix} \quad (26)$$

$$\implies G_t = G_{bak} + G_{exp, t} \quad (27)$$

with

$$W_{bak} = \begin{bmatrix} \mathbb{1}_{N-L-n+1}^\top \\ H_{L+n}(u^d) \\ H_{L+n}(y^d) \end{bmatrix}; \quad W_{exp, t} = \begin{bmatrix} \mathbb{1}_{t-L-n}^\top \\ H_{L+n}(u_{[1, t-1]}) \\ H_{L+n}(y_{[1, t-1]}) \end{bmatrix} \quad (28)$$

$$G_{bak} = W_{bak} K_{bak}^{-1} W_{bak}^\top; \quad G_{exp, t} = W_{exp, t} K_{exp, t}^{-1} W_{exp, t}^\top \quad (29)$$

(27) allows us to build  $G_t$  as the sum of a fixed  $G_{bak}$  and an recursively updated matrix  $G_{exp, t}$  following (22).

#### IV. EXAMPLES AND EXPERIMENTAL RESULTS

We first demonstrate the efficiency of Problem 10 compared to Problem 3 in a simulation experiment, by showing that it produces the same closed-loop trajectory, but requiring only a fraction of the computation time.

We then study a real-system implementation of the reduced-formulation DeePC with exponential forgetting, where we consider two possible options: using a single dataset and using a mosaic two-dataset modification. This experiment highlights the advantage of having a “backup” dataset to circumvent the possible lack of excitation mentioned in Section III-D.

As the real-time testbed, we consider a heat-blower system with two actuators, blower fan and heating resistance, and two outputs given by a flow meter and a thermocouple. This system is non-linear and was shown to need the two-datasets formulation of the DeePC framework; see [31] for more details.

The simulation model required for the validating experiment was obtained by identification, using truncated Volterra series of degree 2.

##### A. Simulation: compute time difference

In order to compare the closed-loop trajectories induced by Problem 10 and Problem 3, we consider a simulated numerical experiment. We formulate the optimal control problem as Problem 4 with  $Q = \mathbb{I}$ ,  $\lambda_\sigma = 10$ ,  $\lambda_g = 1$ . The prediction horizon is set to  $L = 41$ . The initial conditions consist of  $n = 8$  samples, and the sample time is  $T_s = 0.3s$ . The length of the initial dataset takes different values to show the impact on compute time:  $N \in \{246, 348, 464, 696, 928\}$  samples. The length of the updated dataset is kept constant at  $N_o = L + n + 1$ .

Simulations are carried out with MATLAB®2023b, on a laptop equipped with an Intel®Core i7-8750H six-core CPU. The solver for quadprog uses the “active set” method, with default parameters and ‘the initial guess  $x_0 = 0$ . The computational time is estimated using the MATLAB Profiler. Each experiment is run at least 2 times to ensure consistency

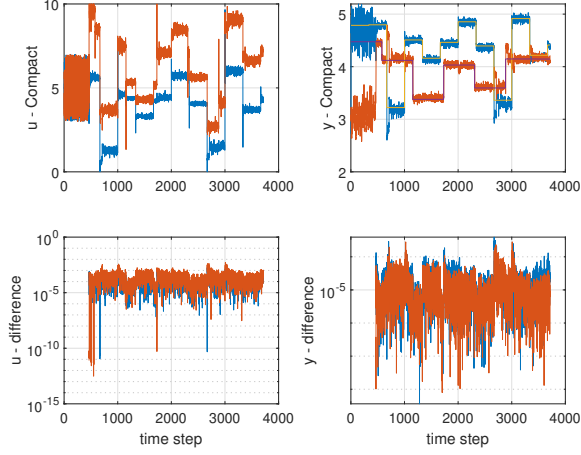


Fig. 1. Input (left) - output (right) trajectory (top) and the absolute difference between formulations (bottom), for a run with  $N = 464$ . Notice the initial input excitation during  $N$  time steps. Thin solid (yellow and purple) line correspond to the reference signal  $y^r$ . The order of magnitude of the difference between inputs due to our reformulation is around  $5 \times 10^{-3}$ , which is negligible in our setup.

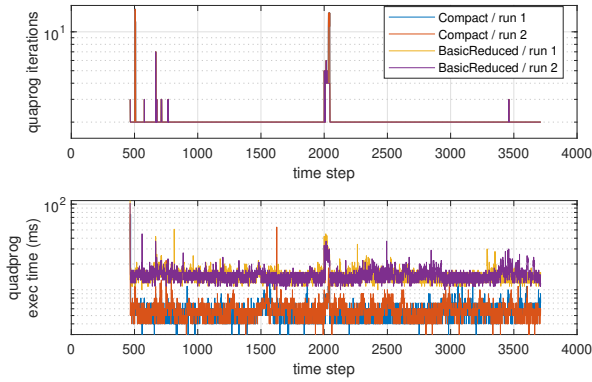


Fig. 2. Computation time and number of iterations performed by `quadprog` at each time step, for  $N = 464$ . Blue and red curves show our reformulation, while the original is in purple and yellow. Our reformulation sometimes takes more iterations than the original due to the numerical behavior of  $G$ , but the computation time is significantly and consistently lower.

of these estimates. In all figures, the references for the two output dimensions are shown by purple and yellow lines.

Fig. 1 shows, the closed-loop trajectory  $y$  and the corresponding command signal  $u$  produced by Problem 10 and the differences with respect to the same signals obtained solving Problem 3. It clearly appears that both methods are essentially equivalent. On the other hand, the computation time of the proposed method is consistently lower through the entire experiment, as shown in Fig. 2 (approximately  $5ms$ , versus  $12ms$  for the original).

We then compare the average computation time for the different dataset sizes  $N$ , with a fixed prediction horizon  $L = 41$  samples, on Fig. 3. As expected, the computation time for Problem 3 increases greatly (seemingly in a polynomial

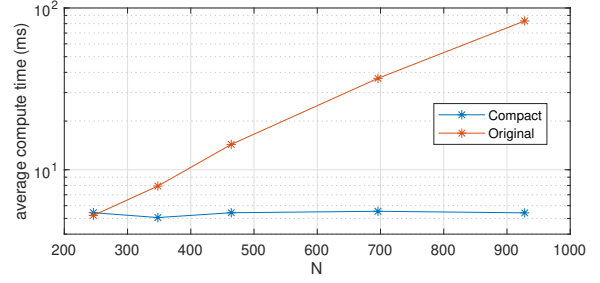


Fig. 3. Average compute time for `quadprog`, for different dataset length  $N$ . Our reformulation is shown on the blue line, while the original is the red curve. As expected, the original formulation takes more and more time to solve as the quantity of data increases, while our reformulation keeps an almost constant (with minor variations) compute time.

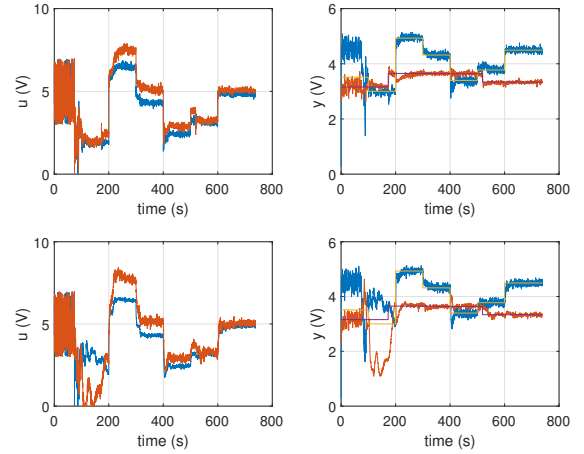


Fig. 4. Input (left) - output (right) trajectory of a real system, with exponential forgetting. Thin solid (yellow and purple) lines correspond to the reference signal  $y^r$ . The presence of a backup dataset (top) clearly improves tracking performance as data can be “flattened” by the Exponential Forgetting (bottom).

fashion) with  $N$ , contrarily to Problem 10. Small variations still incur, due to the fact that compute time is not strictly deterministic: on the one hand, `quadprog`’s algorithm can perform a variable amount of iterations for the same problem size (as seen on Fig. 2 top chart); on the other hand, the speed of each operation performed by a CPU can be affected by several factors (such as background load or CPU temperature; hence variations in Fig. 2 bottom chart).

### B. Real system

We now present an implementation of exponential forgetting on a real system using the two-datasets formulation given in Section III-E.2. Since the original full-size problem formulation does not allow for a straightforward exponential forgetting modification, we consider only the reduced reformulation here.

We set  $\rho = 0.99$ ,  $N = 246$ . We take  $K_{bak} = 20\mathbb{I}$  so that the most recent data is preferred to the backup dataset. We run one experiment with the backup dataset and one without (where  $G_t = G_{exp,t}$ ) to show its effect (see Fig. 4).

It is pretty clear that, while tracking is somewhat achieved with only exponential forgetting, the backup dataset improves performance significantly. We emphasize that adding this supplementary data does not directly impact the computational complexity of solving the OCP.

### C. Discussion

The presented simulation and experimental results illustrate the benefits of the proposed reformulation. However, these benefits come with limits. First, we assumed a specific form of optimization problem, which gives a direction for future works: in particular, we define the regularization based on a positive definite quadratic form  $\|g\|_K$ , but it is not the only way to go (there exist regularizations based on projectors ( $\mathbb{I} - Z^\dagger Z$  with  $Z = [U^\top Y_p^\top]^\top$ ) or  $\ell_1$  norm [33], [25], [2], [34]). Furthermore, if  $h(\|g\|_K)$  is strictly convex, there is no guarantee that  $h(\|\alpha\|_G)$  still is: if  $W$  does not have full row rank, we only have  $K \succ 0 \implies G \succeq 0$ . Finally, despite the globally better compute time, matrix  $G$  has a worse condition number than  $W$  (e.g. with  $K = \mathbb{I}$ ,  $\text{cond}(G) = \text{cond}(W)^2$ ), which can deteriorate numerical accuracy, or require more iterations in optimization algorithms. This also leaves room for improvement.

### V. CONCLUSION

In this paper, we proposed a computationally efficient reformulation for DeePC, based on range space equivalence, that only involves the replacement of two matrices, but relies on problems of much smaller dimension. We proved that the reformulation is equivalent to the original problem, and the reformulation allows for efficient data management through, e.g., exponential forgetting. Experiments validated the proposed method and highlighted the huge computational savings it can bring. We demonstrated the possibility of mixing different data management strategies by coupling a fixed dataset with exponential forgetting. This work opens several perspectives, in particular extending the method to other regularization terms, or addressing the potential conditioning issues.

### REFERENCES

- [1] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, Apr. 2005.
- [2] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annual Reviews in Control*, vol. 52, pp. 42–64, Jan. 2021.
- [3] I. Markovsky, J. C. Willems, P. Rapisarda, and B. L. M. D. Moor, "DATA DRIVEN SIMULATION WITH APPLICATIONS TO SYSTEM IDENTIFICATION," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 970–975, Jan. 2005.
- [4] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, Dec. 2008.
- [5] C. De Persis and P. Tesi, "Formulas for Data-Driven Control: Stabilization, Optimality, and Robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, Mar. 2020.
- [6] J. Berberich and F. Allgöwer, "A trajectory-based framework for data-driven system analysis and control," in *2020 European Control Conference (ECC)*, May 2020, pp. 1365–1370.
- [7] J. Coulson, J. Lygeros, and F. Dörfler, "Data-Enabled Predictive Control: In the Shallows of the DeePC," in *2019 18th European Control Conference (ECC)*, Jun. 2019, pp. 307–312.
- [8] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-Driven Model Predictive Control With Stability and Robustness Guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021.
- [9] M. Lazar and P. C. N. Verheijen, "Offset-free data-driven predictive control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, Dec. 2022, pp. 1099–1104.
- [10] H. Yang and S. Li, "A data-driven predictive controller design based on reduced Hankel matrix," in *2015 10th Asian Control Conference (ASCC)*, May 2015, pp. 1–7.
- [11] S. Baros, C.-Y. Chang, G. E. Colón-Reyes, and A. Bernstein, "Online data-enabled predictive control," *Automatica*, vol. 138, p. 109926, Apr. 2022.
- [12] R. Ou, G. Pan, and T. Faulwasser, "Data-Driven Multiple Shooting for Stochastic Optimal Control," *IEEE Control Systems Letters*, vol. 7, pp. 313–318, 2023.
- [13] J. Coulson, J. Lygeros, and F. Dörfler, "Regularized and Distributionally Robust Data-Enabled Predictive Control," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, Dec. 2019, pp. 2696–2701.
- [14] —, "Distributionally Robust Chance Constrained Data-Enabled Predictive Control," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3289–3304, Jul. 2022.
- [15] Y. Lian, J. Shi, M. Koch, and C. N. Jones, "Adaptive Robust Data-Driven Building Control via Bilevel Reformulation: An Experimental Result," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 6, pp. 2420–2436, Nov. 2023.
- [16] L. Di Natale, Y. Lian, E. T. Maddalena, J. Shi, and C. N. Jones, "Lessons Learned from Data-Driven Building Control Experiments: Contrasting Gaussian Process-based MPC, Bilevel DeePC, and Deep Reinforcement Learning," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, Dec. 2022, pp. 1111–1117.
- [17] 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*, vol. 31, no. 18, pp. 8916–8936, 2021.
- [18] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Data-Enabled Predictive Control for Grid-Connected Power Converters," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, Dec. 2019, pp. 8130–8135.
- [19] D. Bilgic, A. Koch, G. Pan, and T. Faulwasser, "Toward data-driven predictive control of multi-energy distribution systems," *Electric Power Systems Research*, vol. 212, p. 108311, Nov. 2022.
- [20] K. Zhang, Y. Zheng, C. Shang, and Z. Li, "Dimension Reduction for Efficient Data-Enabled Predictive Control," *IEEE Control Systems Letters*, vol. 7, pp. 3277–3282, 2023.
- [21] P. Schmitz, M. Schaller, M. Voigt, and K. Worthmann, "Fast and memory-efficient optimization for large-scale data-driven predictive control," Feb. 2024.
- [22] A. Vahidi-Moghaddam, K. Zhang, Z. Li, and Y. Wang, "Data-Enabled Neighboring Extremal Optimal Control: A Computationally Efficient DeePC," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. Singapore, Singapore: IEEE, Dec. 2023, pp. 4778–4783.
- [23] B. L. M. De Moor, W. Favoreel, and M. Gevers, "SPC: Subspace Predictive Control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004–4009, Jul. 1999.
- [24] J.-W. Van Wingerden, S. P. Mulders, R. Dinkla, T. Oomen, and M. Verhaegen, "Data-enabled predictive control with instrumental variables: The direct equivalence with subspace predictive control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. Cancun, Mexico: IEEE, Dec. 2022, pp. 2111–2116.
- [25] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging Direct and Indirect Data-Driven Control Formulations via Regularizations and Relaxations," *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 883–897, Feb. 2023.
- [26] M. Lazar and P. C. N. Verheijen, "Generalized Data-Driven Predictive Control: Merging Subspace and Hankel Predictors," *Mathematics*, vol. 11, no. 9, p. 2216, Jan. 2023.
- [27] 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*, vol. 67, no. 9, pp. 4406–4421, Sep. 2022.
- [28] A. Martinelli, M. Gargiani, M. Draskovic, and J. Lygeros, "Data-Driven Optimal Control of Affine Systems: A Linear Programming Perspective," *IEEE Control Systems Letters*, vol. 6, pp. 3092–3097, 2022.

- [29] M. Lazar and P. C. N. Verheijen, “Generalized Data-driven Predictive Control,” Apr. 2023.
- [30] A. Goel, A. L. Bruce, and D. S. Bernstein, “Recursive least squares with variable-direction forgetting: Compensating for the loss of persistency [lecture notes],” *IEEE Control Systems Magazine*, vol. 40, no. 4, pp. 80–102, 2020.
- [31] A. Faye-Bédérin, S. Aranovskiy, P. Chauchat, and R. Bourdais, “Maintaining a Relevant Dataset for Data-Driven MPC Using Willems’ Fundamental Lemma Extensions,” in *2023 62nd IEEE Conference on Decision and Control (CDC)*, Dec. 2023, pp. 2584–2589.
- [32] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi, “Willems’ Fundamental Lemma for State-Space Systems and Its Extension to Multiple Datasets,” *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 602–607, Jul. 2020.
- [33] I. Markovsky, L. Huang, and F. Dörfler, “Data-driven control based on behavioral approach: From theory to applications in power systems,” *IEEE Control Syst.*, 2022.
- [34] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, “Robust Data-Enabled Predictive Control: Tractable Formulations and Performance Guarantees,” *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 3163–3170, May 2023.