On the relationship between data-enabled predictive control and subspace predictive control

Felix Fiedler[1] and Sergio Lucia[1]

Abstract— **Data-enabled predictive control (DeePC) is a recently proposed approach that combines system identification, estimation and control in a single optimization problem, for which only recorded input/output data of the examined system is required. The same premise holds for the subspace predictive control (SPC) method in which a multi-step prediction model is identified from the same data as required for DeePC. This model is then used to formulate a similar optimal control problem. In this work we investigate the relationship between DeePC and SPC. Our primary contribution is to show that SPC is equivalent to DeePC in the deterministic case. We also show the equivalence of both methods in a special case for the non-deterministic formulation. We investigate the advantages and shortcomings of DeePC as opposed to SPC with and without measurement noise and illustrate them with a simulation example.**

I. INTRODUCTION

Model predictive control (MPC) is a popular strategy to control multivariable systems with constraints [1]. Traditionally, the main shortcoming of MPC is its computational complexity as it requires the online solution of an optimization problem. Due to advances in computing power, algorithms and efficient implementation strategies, solution time is often not an obstacle for its deployment anymore.

However, MPC still requires a dynamic model of reasonable accuracy to obtain a satisfactory performance. Such model can be obtained from physical modelling, using data-based system identification or a combination of both approaches. Learning from data has recently regained attention although it has been studied for a long time in the field of system identification [2], [3], [4], [5]. Important categories in this vast field are, among others, linear vs. nonlinear [2] and input/output vs. state-space models [3]. A concrete example of interest is the autoregressive model with exogeneous inputs (ARX) [5], a popular linear input/output approach. Typically, ARX models predict the next output of the system and can be evaluated repeatedly to compute finite future trajectories as required for an MPC formulation. Subspace predictive control (SPC) [6] follows a similar approach, where a multi-step ahead prediction model is used to compute the finite future trajectory in a single step. An extensive overview on the method is given in [7] and SPC is still an active field of research with the work in [8] investigating stability and horizon tuning.

In recent years, a new trend in data-based control seeks to integrate the traditional sequential system identification and control in a unified approach. From the machine learning community, reinforcement-learning is a popular approach following this paradigm [9]. However, reinforcement-learning comes with its own challenges such as the demand for large quantities of data and produces highly variable outcomes [10]. Another approach to unify identification and control is the newly proposed data-enabled predictive control (DeePC) [11] algorithm. The approach describes a simple configuration of a predictive control problem which operates directly on matrices of collected input/output data. Furthermore, the required data has to satisfy only moderate requirements and can often be taken directly from a running process.

The initial work on DeePC [11] has thus sparked considerable interest in the control community with several extensions and modifications. The authors in [12] investigate DeePC with respect to stability and robustness and propose an adaptation to guarantee these properties. Distributionally robust DeePC is presented in [13], where it is assumed that data is sampled from a distribution of possible systems. In [14], DeePC is combined with an extended Kalman Filter to improve the performance in the case of noisy measurements.

However, DeePC remains in its nature a linear MPC scheme and is based on data which could also be used for sequential identification and control approaches such as SPC. The authors in [11] state that an equivalent classical MPC formulation exists for their proposed DeePC scheme. This equivalence is based on theorizing about the existence of an unknown system parameterization. But if such a parameterization could be known, what would be the advantages and potential shortcomings of the DeePC approach, especially in the linear setting, for which DeePC is derived?

The main contribution of this work is to help answer this question. In particular we investigate the relationship between DeePC and SPC. The performance of SPC and DeePC has recently been empirically compared in [15]. From the obtained similar performance the authors conclude that there are some similarities between the two formulations. In this work, we prove that in the linear deterministic case, DeePC is equivalent to SPC. The SPC multi-step prediction model can be determined from the exact same data that DeePC requires to operate. This exact equivalence has, to the best of our knowledge, not yet been established in the literature. For the non-deterministic case, DeePC requires some adaptations to deal with the noise corrupted data. We present a minor modification of DeePC and as a second

contribution prove that in a special case this formulation is still equivalent to SPC. Finally, we compare both controllers numerically on a linear model with additive Gaussian noise. We include an analysis of the effect that the amount of available data and the regularization term of DeePC have on the closed-loop performance.

The remainder of this paper is structured as follows. In Section II we present the problem setup and review the data-enabled predictive control algorithm. We then present in Section III the subspace predictive control (SPC) algorithm and prove equivalence to the DeePC method in Theorem 1. We proceed to analyse the case of a non-deterministic linear system in Section IV. We present a minor modification of the original DeePC problem and prove equivalence to SPC in a special case in Theorem 2. Finally, we present a numerical comparison of both controllers in the general case in Section V. We finish this work with conluding remarks in Section VI.

II. DATA-ENABLED PREDICTIVE CONTROL

We investigate an unknown discrete-time LTI dynamic system represented in state-space form:

$$x_{k+1} = Ax_k + Bu_k, \tag{1a}$$

$$y_k = Cx_k + Du_k, \tag{1b}$$

with $x \in \mathbb{R}^n$ (states), $u \in \mathbb{R}^m$ (inputs), $y \in \mathbb{R}^p$ (measurements) and system matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$. We assume that the system with n states is given in its minimal representation.

From this unknown system, we collect T sequences of input/output data of length $L = T_{\text{ini}} + N$, where T_{ini} denotes the length of the initialization sequence and N the length of the prediction horizon. The collected data is arranged in matrices, such that:

$$U_L = \left[u_L^1, u_L^2, \ldots, u_L^T \right], \; Y_L = \left[y_L^1, y_L^2, \ldots, y_L^T \right], \tag{2}$$

where $u_L^1 = \left[(u_1^1)^\mathsf{T}, \ldots, (u_L^1)^\mathsf{T} \right]^\mathsf{T} \in \mathbb{R}^{Lm}$ and $y_L^1 = \left[(y_1^1)^\mathsf{T}, \ldots, (y_L^1)^\mathsf{T} \right]^\mathsf{T} \in \mathbb{R}^{Lp}$. These data matrices may or may not coincide with a Hankel or Page matrix (see [13]). We also assume that the sequences were collected starting from T unknown initial states x_1:

$$X_1 = \left[x_1^1, x_1^2, \ldots, x_1^T \right], \tag{3}$$

where $X_1 \in \mathbb{R}^{n \times T}$. We divide the input/output data in (2) in two parts, such that:

$$U_L = \begin{bmatrix} U_{T_{\text{ini}}} \\ U_N \end{bmatrix}, \; Y_L = \begin{bmatrix} Y_{T_{\text{ini}}} \\ Y_N \end{bmatrix}. \tag{4}$$

To clarify this notation, we now have $U_{T_{\text{ini}}} \in \mathbb{R}^{T_{\text{ini}}m \times T}$, $U_N \in \mathbb{R}^{Nm \times T}$, $Y_{T_{\text{ini}}} \in \mathbb{R}^{T_{\text{ini}}p \times T}$ and $Y_N \in \mathbb{R}^{Np \times T}$.

Definition 1 ([11]): With L, T such that $T \geq Lm$, the signals comprising matrix U_L are persistently exciting of order L if matrix U_L has full rank.

Definition 2 ([11]): We denote with $l(A, B, C, D)$ the lag of system (1). It is defined as the smallest integer l for which

the observability matrix:

$$O_l(A, C) := \left(C, \; CA, \; \ldots, CA^{l-1} \right), \tag{5}$$

has full rank.

Assumption 1: The number of recorded sequences is $T \geq Lm + n$. Matrix U_L is persistently exciting of order L according to Definition 1.

Assumption 2: The initialization sequence is $T_{\text{ini}} > l(A, B, C, D)$, with l according to Definition 2.

Assumption 3: The underlying matrix of initial states (3) has full rank, i.e. rank$(X_1) = n$ and the stacked matrix $[X_1^\mathsf{T}, U_L^\mathsf{T}]^\mathsf{T}$ has full rank.

Subsequently, we state a variant (see Remark 1) of the fundamental lemma of behavioral systems theory [16].

Lemma 1: Let (4) be input/output data of a deterministic system in the form of (1) which satisfies Assumption 1, Assumption 2 and Assumption 3. Any sequence $y_{T_{\text{ini}}} = \left[y_1^\mathsf{T}, \ldots, y_{T_{\text{ini}}}^\mathsf{T} \right]^\mathsf{T}$, $y_N = \left[y_{T_{\text{ini}}+1}^\mathsf{T}, \ldots, y_{T_{\text{ini}}+N}^\mathsf{T} \right]^\mathsf{T}$, $u_{T_{\text{ini}}} = \left[u_1^\mathsf{T}, \ldots, u_{T_{\text{ini}}}^\mathsf{T} \right]^\mathsf{T}$ and $u_N = \left[u_{T_{\text{ini}}+1}^\mathsf{T}, \ldots, u_{T_{\text{ini}}+N}^\mathsf{T} \right]^\mathsf{T}$ is a trajectory of system (1) if and only if, there exists a $g \in \mathbb{R}^T$, such that:

$$\begin{bmatrix} Y_{T_{\text{ini}}} \\ U_{T_{\text{ini}}} \\ U_N \\ Y_N \end{bmatrix} g = \begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ u_N \\ y_N \end{bmatrix}. \tag{6}$$

Remark 1: Note that the original statement of the fundamental lemma in [16] requires the matrices in (2) to be Hankel matrices. In [13] it is shown that (6) also holds for Page matrices which can be extended to show that arbitrary sequences are sufficient as long as the space spanned by the initial states of these sequences has full rank (see Assumption 3). The full proof of Lemma 1 is omitted here for brevity.

Based on the fundamental Lemma 1, the authors in [11] proposed the elegant data-enabled predictive control scheme (DeePC), where the optimal control policy is obtained as the solution of:

$$\begin{aligned} \min_{g, u_N, y_N} \quad & f(g, u_N, y_N) \\ \text{s.t.} \quad & (6) \\ & u_k \in \mathbb{U} \; \forall k \in \{1, \ldots, N\}, \\ & y_k \in \mathbb{Y} \; \forall k \in \{1, \ldots, N\}, \end{aligned} \tag{7}$$

with arbitrary objective function $f(g, u_N, y_N)$. Within the scope of this work, we only investigate the classical regulation task:

$$f(g, u_N, y_N) = \sum_{k=1}^N \left(\|y_k\|_Q^2 + \|u_k\|_R^2 \right), \tag{8}$$

where $\|y_k\|_Q^2 = y_k^\mathsf{T} Q y_K$. We require that Q and R are positive definite.

223

III. SUBSPACE PREDICTIVE CONTROL

We start this section by stating a well known data-based approach for linear system identification: The auto-regressive model with external inputs (ARX) [5], where

$$y_{T_{\text{ini}}+1} = \bar{a}_1 y_1 + \cdots + \bar{a}_{T_{\text{ini}}} y_{T_{\text{ini}}} + \bar{b}_1 u_1 + \cdots + \bar{b}_{T_{\text{ini}}} u_{T_{\text{ini}}}. \quad (9)$$

Similarly as in Lemma 1, the equation can only hold if $T_{\text{ini}} > l(A,B,C,D)$, according to Definition 2 as shown in [17].

To identify the parameters

$$\bar{P} = \begin{bmatrix} \bar{a}_1, & \ldots, & \bar{a}_{T_{\text{ini}}}, & \bar{b}_1, & \ldots, & \bar{b}_{T_{\text{ini}}} \end{bmatrix} \quad (10)$$

based on the collected data in (2), one would typically solve the least-squares problem:

$$\min_{\bar{P}} \quad \left\| \bar{P} \begin{bmatrix} Y_{T_{\text{ini}}} \\ U_{T_{\text{ini}}} \end{bmatrix} - Y_+ \right\|_F^2, \quad (11)$$

where $\|\cdot\|_F$ denotes the Frobenius-norm. The matrix Y_+ represents the first p rows of Y_N corresponding to the temporal elements of the output sequences at $T_{\text{ini}} + 1$. Reminiscent of the ARX model in (9), the SPC approach [6] uses the collected data (2) to directly identify a linear multi-step ahead predictor:

$$y_N = a_1 y_1 + \cdots + a_{T_{\text{ini}}} y_{T_{\text{ini}}} \quad (12)$$
$$b_1 u_1 + \cdots + b_{T_{\text{ini}}+N} u_{T_{\text{ini}}+N},$$

where the parameters are again summarized as

$$P = [a_1, \ldots, a_{T_{\text{ini}}}, b_1, \ldots, b_{T_{\text{ini}}+N}]. \quad (13)$$

Lemma 2: Let (4) be input/output data of a deterministic system in the form of (1) which satisfies Assumption 1, Assumption 2 and Assumption 3. Let matrix $P^* \in \mathbb{R}^{Np \times Nm + T_{\text{ini}}(m+p)}$ be the solution of the least squares problem

$$P^* = \arg \min_P \left\| P \underbrace{\begin{bmatrix} Y_{T_{\text{ini}}} \\ U_{T_{\text{ini}}} \\ U_N \end{bmatrix}}_{M} - Y_N \right\|_F^2, \quad (14)$$

which is expressed explicitly in terms of the Moore-Penrose inverse (denoted with the superscript †) as:

$$P^* = Y_N M^\dagger. \quad (15)$$

Any sequence $u_{T_{\text{ini}}}, u_N, y_{T_{\text{ini}}}, y_N$ is a trajectory of system (1) if

$$y_N = P^* \begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ u_N \end{bmatrix}. \quad (16)$$

Proof: The proof is shown in the appendix. ∎

Lemma 2 allows to state the subspace predictive control problem (SPC) as

$$\min_{u_N, y_N} \quad \sum_{k=1}^{N} \left(\|y_k\|_Q^2 + \|u_k\|_R^2 \right)$$
$$\text{s.t.} \quad (16) \quad (17)$$
$$u_k \in \mathbb{U}, \ \forall k \in \{1, \ldots, N\},$$
$$y_k \in \mathbb{Y}, \ \forall k \in \{1, \ldots, N\}.$$

Remark 2: In the original work on SPC [6] the authors choose $T_{\text{ini}} = N$. For the comparison with DeePC we follow the SPC formulation from [8] with dedicated parameter T_{ini}.

A. Equivalence of DeePC and SPC

In the following we present our main contribution, a theorem stating that subspace predictive control (17) is an equivalent representation of the DeePC (7) method.

Theorem 1: Let U_L, Y_L in the form of (4), and $u_{T_{\text{ini}}}, y_{T_{\text{ini}}}$ be recorded from the linear system (1). The recorded data satisfies Assumption 1, Assumption 2 and Assumption 3. The optimal control policy u_N^* and optimal output trajectory y_N^* as obtained from the solution of the DeePC problem (7) is equal to the optimal control policy u_N^* and optimal output trajectory y_N^* as obtained from the solution of the SPC problem (17).

Proof: We modify problem (7) and eliminate the variable g from the formulation. First, we split (6) into:

$$\begin{bmatrix} Y_{T_{\text{ini}}} \\ U_{T_{\text{ini}}} \\ U_N \end{bmatrix} g = \begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ u_N \end{bmatrix}, \quad (18)$$

and

$$y_N = Y_N g. \quad (19)$$

From (18) we can explicitly compute g^* as:

$$g^* = M^\dagger b + \hat{g} \quad \forall \hat{g} \in \ker(M), \quad (20)$$

where we denote with $\ker(M)$ the null-space of matrix M. Inserting (20) in (19) yields:

$$y_N = Y_N g^*, \quad (21)$$
$$\overset{(20)}{\Longleftrightarrow} y_N = Y_N(M^\dagger b + \hat{g}). \quad (22)$$

We then need to show that $\ker(M) \subseteq \ker(Y_N)$, such that

$$Y_N \hat{g} = 0 \quad \forall \hat{g} \in \ker(M). \quad (23)$$

The relationship $\ker(M) \subseteq \ker(Y_N)$ holds because P^* as obtained from (15), according to Lemma 2, satisfies (16) and thus:

$$Y_N = P^* M. \quad (24)$$

For any $x \in \ker(M)$ we have:

$$Y_N x = P^* M x = 0, \quad (25)$$

which means $x \in \ker(Y_N)$ and consequently $\ker(M) \subseteq \ker(Y_N)$. Finally, we obtain from (22):

$$y_N = Y_N(M^\dagger b) \quad (26)$$
$$= P^* b. \quad (27)$$

We have thus eliminated g from the formulation of (7), where the constraint (6) now reads:

$$y_N = P^* \begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ u_N \end{bmatrix}. \quad (28)$$

This yields the presented SPC formulation (17) and therefore proves that (7) and (17) have the same solution. ∎

As a consequence of the equivalence between problem (7) and (17), we argue that the DeePC optimization problem implicitly estimates the regressor (15) at each control iteration. Since the underlying data-matrices (2) are collected prior to the control application and are unchanged afterwards, there appears to be no reason for this repeated estimation in the deterministic case. The DeePC optimization problem (7) requires $T + (m+p)N$ optimization variables with $T \geq (T_{\text{ini}} + N)m + n$ (see Assumption 1) and $(m+p)(T_{\text{ini}} + N)$ equality constraints. The SPC formulation (17) has only $(m+p)N$ optimization variables and pN equality constraints. The increased computational cost of DeePC can be also seen in the numerical example presented in Section V.

IV. NON-DETERMINISTIC CASE

We introduce the following linear system subject to zero-mean Gaussian noise $w_k \sim \mathcal{N}(0, \sigma_w^2 I)$:

$$x_{k+1} = Ax_k + Bu_k, \tag{29a}$$
$$y_k = Cx_k + Du_k + w_k. \tag{29b}$$

As for the deterministic case, data is collected, now resulting in the matrices:

$$U_L = \begin{bmatrix} U_{T_{\text{ini}}} \\ U_N \end{bmatrix}, \quad \tilde{Y}_L = \begin{bmatrix} \tilde{Y}_{T_{\text{ini}}} \\ \tilde{Y}_N \end{bmatrix}, \quad \tilde{M} = \begin{bmatrix} \tilde{Y}_{T_{\text{ini}}} \\ U_{T_{\text{ini}}} \\ U_N \end{bmatrix}.$$

A. Non-deterministic DeePC

In the non-deterministic setting, the DeePC formulation (7) must be adapted because constraint (6) can generally not be satisfied anymore. These modifications were already proposed in the original work on DeePC [11]. Following the authors in [12], we use ℓ_2-norms instead of ℓ_1-norms to penalize the slack variables in the cost function. We furthermore propose to add an additional slack variable σ_u which, similarly to σ_y, is motivated by additive input noise. This yields the following adapted DeePC formulation:

$$\min_{g, u_N, y_N, \sigma_y, \sigma_u} \quad \sum_{k=1}^{N} \left(\|y_k\|_Q^2 + \|u_k\|_R^2 \right)$$
$$+ \lambda_g \|g\|_2^2 + \lambda_{\sigma_y} \|\sigma_y\|_2^2 + \lambda_{\sigma_u} \|\sigma_u\|_2^2$$
$$\text{s.t.} \quad \begin{bmatrix} \tilde{Y}_{T_{\text{ini}}} \\ U_{T_{\text{ini}}} \\ U_N \\ \tilde{Y}_N \end{bmatrix} g = \begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ u_N \\ y_N \end{bmatrix} + \begin{bmatrix} \sigma_y \\ \sigma_u \\ 0 \\ 0 \end{bmatrix}, \tag{30}$$
$$u_k \in \mathbb{U} \; \forall k \in \{1, \dots, N\},$$
$$y_k \in \mathbb{Y} \; \forall k \in \{1, \dots, N\}.$$

B. Non-deterministic SPC

Subspace predictive control in the non-deterministic setting requires no modifications for the identification step:

$$\tilde{P}^* = \tilde{Y}_N \tilde{M}^\dagger. \tag{31}$$

The SPC optimization problem is adapted to account for noisy initial conditions with slack variables σ_y and σ_u.

$$\min_{u_N, y_N, \sigma_y, \sigma_u} \quad \sum_{k=1}^{N} \left(\|y_k\|_Q^2 + \|u_k\|_R^2 \right) + \lambda_{\sigma_y} \|\sigma_y\|_2^2 + \lambda_{\sigma_u} \|\sigma_u\|_2^2$$
$$\text{s.t.} \quad y_N = \tilde{P}^* \left(\begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ u_{N,} \end{bmatrix} + \begin{bmatrix} \sigma_y \\ \sigma_u \\ 0, \end{bmatrix} \right) \tag{32}$$
$$u_k \in \mathbb{U}, \; \forall k \in \{1, \dots, N\},$$
$$y_k \in \mathbb{Y}, \; \forall k \in \{1, \dots, N\}.$$

The main difference in the non-deterministic setting is that $\tilde{P}^*$ now cannot exactly map $\tilde{M}$ onto $\tilde{Y}_N$. However, due to the underlying least-squares formulation (14), the obtained regressor has favorable statistical properties in this setting, such as being unbiased [18].

C. Equivalence in the non-deterministic case

Equivalence of the adapted DeePC problem (30) and SPC problem (32) in the non-deterministic case cannot generally be established. There are, however, special cases in which both methods yield the exact same optimal control policy u_N^* and optimal output trajectory y_N^*. Subsequently, we present a theorem to show equivalence in such a special case. For this purpose, we reformulate the adapted DeePC problem (30) and SPC problem (32). We stack the optimization variables $v = [\sigma_y, \ \sigma_u, \ u_N]$ and introduce the block-diagonal weighting matrices:

$$\tilde{Q} = \begin{bmatrix} Q & & \\ & \ddots & \\ & & Q \end{bmatrix}, \tilde{R} = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix}, V = \begin{bmatrix} \lambda_{\sigma_y} & & \\ & \lambda_{\sigma_u} & \\ & & \tilde{R} \end{bmatrix}.$$

We then drop the inequality constraints in (30) and (34). This allows to state the unconstrained version of the adapted DeePC problem (30) as:

$$\min_{g, v, y_N} \quad \frac{1}{2} g^\mathsf{T}(\lambda_g I) g + \frac{1}{2} y_N^\mathsf{T} \tilde{Q} y_N + \frac{1}{2} v^\mathsf{T} V v$$
$$\text{s.t.} \quad y_N = \tilde{Y}_N g$$
$$\underbrace{\begin{bmatrix} \tilde{Y}_{T_{\text{ini}}} \\ U_{T_{\text{ini}}} \\ U_N \end{bmatrix}}_{\tilde{M}} g = \underbrace{\begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ 0 \end{bmatrix}}_{b} + \underbrace{\begin{bmatrix} \sigma_y \\ \sigma_u \\ u_N \end{bmatrix}}_{v}, \tag{33}$$

and the unconstrained version of the SPC problem (32) as:

$$\min_{v, y_N} \quad \frac{1}{2} y_N^\mathsf{T} \tilde{Q} y_N + \frac{1}{2} v^\mathsf{T} V^\mathsf{T} v$$
$$\text{s.t.} \quad y_N = \tilde{P} \left(\underbrace{\begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ 0 \end{bmatrix}}_{b} + \underbrace{\begin{bmatrix} \sigma_y \\ \sigma_u \\ u_N \end{bmatrix}}_{v} \right). \tag{34}$$

Dropping the inequality constraints in (33) and (34) has been considered previously for DeePC in [15] and SPC in [8] as it allows for explicit solutions of (33) and (34) which are preferable for very fast control applications. Below, we state these explicit solutions in our notation.

225

Result 1: The optimal value v^* for (33) is computed explicitly as:

$$v^* = \tilde{M}(\lambda_g I + \tilde{M}^\mathsf{T} V \tilde{M} + Y_N^\mathsf{T} \tilde{Q} Y_N)^{-1} \tilde{M}^\mathsf{T} V b - b. \tag{35}$$

Proof: The proof is shown in the appendix. ∎

Result 2: The optimal value v^* for (34) is computed explicitly as:

$$v^* = -(V + \tilde{P}^\mathsf{T} Q \tilde{P})^{-1} \tilde{P}^\mathsf{T} \tilde{Q} \tilde{P} b. \tag{36}$$

Proof: The proof is shown in the appendix. ∎

We use Result 1 and Result 2 to establish equivalence of the DeePC and SPC solution v^* under certain conditions.

Theorem 2: Let $T = Lm + T_{\text{ini}} p$ and Assumption 1, Assumption 2 and Assumption 3 hold. Let $M \in \mathbb{R}^{Lm+T_{\text{ini}}p \times T}$ and $Y_N \in \mathbb{R}^{Np \times T}$ have full rank. The unconstrained DeePC problem (33) and unconstrained SPC problem (34) yield the same optimal solution v^*, if $\lambda_g = 0$.

Proof: We reformulate (36) by applying the following identity [19]:

$$(\Lambda + \Gamma)^{-1} = \Lambda^{-1} - (\Lambda + \Gamma)^{-1} \Gamma \Lambda^{-1} \tag{37}$$

with

$$\Lambda = \tilde{P}^\mathsf{T} \tilde{Q} \tilde{P}, \quad \Gamma = V. \tag{38}$$

Note that (37) only holds if Λ and $\Lambda + \Gamma$ are non-singular. According to the assumptions, we have that $\tilde{M}$ and $\tilde{Y}_N$ have full rank and $\tilde{M}$ is quadratic, such that:

$$\tilde{P}^* = Y_N \tilde{M}^\dagger = \tilde{Y}_N \tilde{M}^{-1} \tag{39}$$

also has full rank. By definition Q, R and therefore $\tilde{Q}$ and V are positive definite. Therefore we have that $\tilde{P}^\mathsf{T} \tilde{Q} \tilde{P}$ and $\tilde{P}^\mathsf{T} \tilde{Q} \tilde{P} + V$ are positive definite. Identity (37) can thus be applied to the explicit solution of the SPC problem (36) and yields:

$$\begin{aligned} v^* &= \left[-(\tilde{P}^\mathsf{T} \tilde{Q} \tilde{P})^{-1} + (V + \tilde{P}^\mathsf{T} \tilde{Q} \tilde{P})^{-1} V (\tilde{P}^\mathsf{T} \tilde{Q} \tilde{P})^{-1} \right] \tilde{P}^\mathsf{T} \tilde{Q} \tilde{P} b \\ &= (V + \tilde{P}^\mathsf{T} \tilde{Q} \tilde{P})^{-1} V b - b. \end{aligned} \tag{40}$$

We substitute (39) in (40):

$$\begin{aligned} v^* &= \left(V + (\tilde{M}^{-1})^\mathsf{T} \tilde{Y}_N^\mathsf{T} \tilde{Q} \tilde{Y}_N \tilde{M}^{-1} \right)^{-1} V b - b \\ &= \left((\tilde{M}^{-1})^\mathsf{T} (\tilde{M}^\mathsf{T} V \tilde{M} + \tilde{Y}_N^\mathsf{T} \tilde{Q} \tilde{Y}_N) (\tilde{M}^{-1}) \right)^{-1} V b - b, \end{aligned} \tag{41}$$

and computing the outer inverse, we obtain

$$v^* = \tilde{M} \left(\tilde{M}^\mathsf{T} V \tilde{M} + \tilde{Y}_N^\mathsf{T} \tilde{Q} \tilde{Y}_N \right)^{-1} \tilde{M}^\mathsf{T} V b - b. \tag{42}$$

This means that if $\lambda_g = 0$ as required in the theorem, the explicit solution (35) of the unconstrained adapted DeePC problem (30) is equal to (42) and therefore equal to the explicit solution (36) of the unconstrained SPC problem (32). ∎

Clearly the equivalence established in Theorem 2 holds only under strong assumptions (i.e. $T = Lm + T_{\text{ini}} p$ and $\lambda_g = 0$) and is shown here only for the unconstrained formulation of DeePC (33) and SPC (34). Furthermore, the presented DeePC formulation in (30) and (33) deviates from formulations in previous works through the addition of the slack variable σ_u. In future work, we therefore aim to investigate
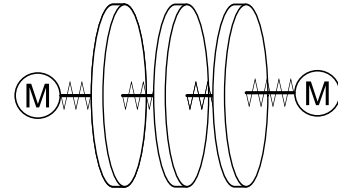

Fig. 1. Triple-mass-spring system (rotating) with two stepper motors as inputs (motor angle). Disc angles can be measured.

the constrained case which would also allow to enforce $\sigma_u = 0$.

Bearing in mind the limitations of Theorem 2, we still argue that it shows that DeePC and SPC are very closely related, even in the non-deterministic formulation. This close relationship also holds in practice as shown in [15] and will be further illustrated in Section V. The main reason for deviations between the DeePC and SPC solution in the general case is due to the regularization of the decision variable g in DeePC with $\lambda_g > 0$. It becomes clear, by inspecting (35), that this regularization is required in the general case, as $\tilde{M}^\mathsf{T} V \tilde{M} + \tilde{Y}_N^\mathsf{T} \tilde{Q} \tilde{Y}_N$ is singular for $T > \max(Lm + T_{\text{ini}} p, \ Np)$. In Section V we numerically investigate the effect of λ_g on the obtained solution.

V. SIMULATION STUDY

In this section, we present numerical evidence of our findings. We investigate a simple LTI system in the form of (1) in two scenarios: with and without additive Gaussian measurement noise. The investigated system, displayed in Figure 1, is a triple-mass-spring system (rotating discs) with two stepper motors ($m = 2$) attached to the outermost discs via additional springs. These disc angles are the measured output of the system ($p = 3$). The system has a total of $n = 8$ states. All materials to reproduce the results in this section (as well as the investigated system) are available in our accompanying repository[1]. According to Definition 2 we determine the system lag $l = 2$. To satisfy Assumption 2, we choose $T_{\text{ini}} = 4$. The prediction horizon is chosen as $N = 40$, which represents 4s. This means we must collect a minimum of $T > Lm + n = 96$ sequences for our data matrices in (2) to satisfy Assumption 1. Note that data is collected from independent experiments with random initial states fulfilling Assumption 3. The regulation objective is parameterized with $Q = I$ and $R = 0.1I$ and we constrain the inputs to $\mathbb{U} = \{-0.7, 0.7\}$. All controller variants are implemented using CasADi [20] with IPOPT [21].

A. Deterministic system

We first consider the described system without measurement noise and compare the deterministic DeePC algorithm (7) with SPC (17). The results are obtained with data from $T = 150$ captured sequences. In the deterministic setting, we have shown in Theorem 1 that both methods yield identical solutions. In Figure 2 we can see that the identical behavior also appears in practice. We present here an exemplary

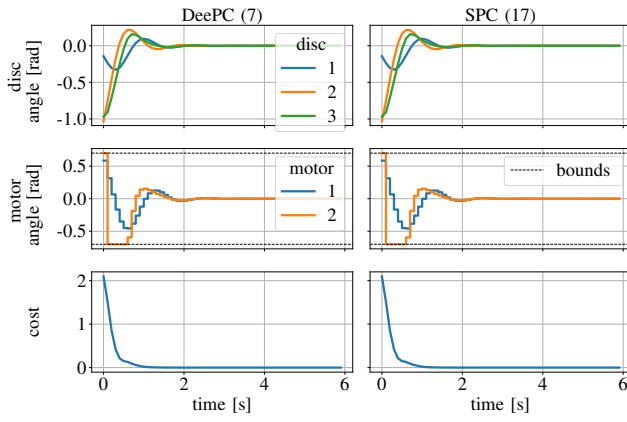[1] https://github.com/4flixt/DeePC_Perspective

Fig. 2. Closed-loop trajectories of the mass-spring-system from Figure 1. Regulation after some initial excitation. Comparison of DeePC (7) and SPC (17) with deterministic data.
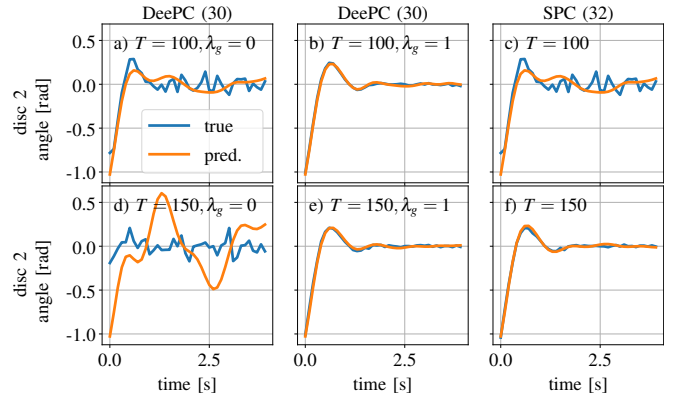

Fig. 3. Open-loop trajectories for the non-deterministic case with DeePC (30) vs. SPC (32). Comparison of predicted vs. true system response (when subjected to the optimal control trajectory). Only disc 2 (see Figure 2) is displayed for clarity.

TABLE I

CLOSED-LOOP TRAJECTORIES FOR THE NON-DETERMINISTIC CASE
WITH DEEPC (30) VS. SPC (32).

		Number of sequences T			Ref.
		100	150	200	
DeePC	cost (43)	5.924	5.674	5.662	5.617
($\lambda_g = 1$)	comp. time [ms]	25.272	39.419	59.291	-
SPC	cost (43)	5.810	5.64	5.632	5.617
	comp. time [ms]	17.30	16.25	16.13	-

closed-loop trajectory for both methods with random (but identical) excitation phase to obtain $y_{T_{\mathrm{ini}}}$ and $u_{T_{\mathrm{ini}}}$. To further quantify the similarity, we compare the cumulated cost:

$$c = \sum_{k=1}^{N_{\mathrm{sim}}} \left(y_k^\mathsf{T} Q y_k + u_k^\mathsf{T} R u_k \right). \qquad (43)$$

For the results in Figure 2 the cumulative cost differs between both controller variants on the order of 10^{-11} and equals to 5.617, which will serve as a benchmark for the non-deterministic case.

B. Non-deterministic system

In this subsection we investigate the previously described LTI system in the form of (29), where we choose $\sigma_w = 10^{-2}$ in all experiments. We choose $\lambda_y = 10^4$ and $\lambda_u = 10^4$ for non-deterministic formulations of DeePC (30) and SPC (32).

We want to compare the performance of both approaches and study the effect of the tuning parameter λ_g and of the number of recorded sequences T on the cost, according to (43), and computation time.

In Figure 3 we present predicted optimal trajectories obtained with (30) and (32) for the same initial input/output data $\tilde{y}_{T_{\mathrm{ini}}}$ and $u_{T_{\mathrm{ini}}}$ (noise disturbed). These trajectories (denoted as pred. in Figure 2) are compared with the true system response resulting from the sequence of optimal inputs. We vary the number of recorded sequences $T = \{100, 150\}$ and change the value of $\lambda_g = \{0, 1\}$. For a more concise representation we only showcase the output of the second disc angle. Notice that $T = 100$ barely exceeds $T = 96$, the lower bound to satisfy Assumption 1. With $T = 100$ and $\lambda_g = 0$ we also satisfy the assumptions for Theorem 2, meaning that we expect identical solutions from the unconstrained DeePC problem (33) and the SPC problem (34). This equivalence can also be observed for the constrained DeePC and SPC formulations in Figure 3 when comparing case a) and c). The open loop-cost of these cases differs on the order of 10^{-13}. Another observation from Figure 3 case d) is the unsatisfactory performance of DeePC, which stems from the

fact that with $\lambda_g = 0$ and $T > \max(Lm + T_{\mathrm{ini}}p, \, Np) = 120$ a singular matrix arises in the computation of v^*. Good performance with SPC can be obtained with $T = 150$ as seen in case f) or with DeePC for $T = 150$ and $\lambda_g = 1$ as shown in case e).

A further comparison is presented in Table I, where we analyze closed-loop trajectories obtained with DeePC (30), with $\lambda_g = 1$, and SPC (32). All results in this table are averages over ten simulation experiments, with independently sampled data matrices and measurement noise.

We take two main conclusions from the presented data in Table I. First, we see that SPC outperforms DeePC both in terms of overall cost and computation time in all scenarios. While the advantage of SPC with respect to the cost is minor, the difference in computation time is significant. Especially for the case $T = 150$, which seems to be a good compromise between performance and cost for DeePC and SPC, we notice a significant increase in computation time. Note that we compare only the online CPU time. The computation of the pseudo inverse is excluded, as it is performed offline and only once. As a second conclusion, we notice that even with $T = 100$, DeePC is outperformed by SPC. This is counter intuitive in comparison to Figure 3 case b) and c), where open-loop trajectories seem better with DeePC. However, open-loop predictions are not the same as close-loop control and we find that SPC shows less oscillatory behavior around the origin (not shown here), thus leading to a slightly lower closed-loop cost.

227

Note that we also investigated DeePC (30), with $\lambda_g \in \{0.1, 10\}$ and $T = 150$. Compared to the displayed case $\lambda_g = 1$ and $T = 150$ in Table I, the different parameters for λ_g lead to increased closed-loop costs, according to (43), of 5.707 and 5.984. As a conclusion, we found that $\lambda_g = 1$ is an appropriate choice for the presented comparison.

VI. CONCLUSIONS

In this work we investigated the relationship between data-enabled predictive control (DeePC) and subspace predictive control (SPC). Both methods require for their formulation the exact same data matrices, satisfying identical requirements. With DeePC, these matrices are directly incorporated in the optimal control problem, whereas SPC requires the data to identify a multi-step prediction model. We show that in the case of deterministic data, both formulations are equivalent. From the equivalence, we reason that DeePC implicitly estimates the same multi-step ahead prediction model at each iteration, thus adding significant additional computational cost to its online application. In particular, the online computational cost of DeePC grows with the number of collected data sequences.

For the non-deterministic case, the exact equivalence between DeePC and SPC does not hold in the general case. We propose a minor modification of DeePC and prove that this formulation is equivalent to SPC in the unconstrained case under special conditions. Even though the presented equivalence only holds under strong assumptions, it illustrates the close relationship between both methods.

In simulation studies with an exemplary LTI system with and without additive Gaussian noise, we showcase that the derived equivalences also hold in practice. Furthermore, we investigate the non-deterministic DeePC and SPC formulation with respect to the number of recorded data sequences and the regularization term for DeePC. We find that SPC outperforms DeePC in the investigated cases with minor improvements of the closed-loop cost and significant improvements in the online computation time.

In future work, we seek to further investigate the relationship between both methods and aim to extend the equivalence of the stochastic formulations to the constrained case. We also seek to compare both methods for nonlinear or time-varying systems where the implicit re-estimation of DeePC could be beneficial.

APPENDIX

A. Proof of Lemma 2

Proof: With the observability matrix $O_i(A,C)$ and the Toeplitz matrix $H_i(A,B,C,D)$

$$O_i = \begin{bmatrix} C, \\ CA, \\ \vdots \\ CA^{i-1} \end{bmatrix}, \quad H_i = \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & & \ddots & \\ CA^{i-2}B & CA^{i-3}B & \dots & D \end{bmatrix}, \quad (44)$$

if the initial state x_1 is known, we can obtain $y_{T_{\text{ini}}}$ as:

$$y_{T_{\text{ini}}} = O_{T_{\text{ini}}} x_1 + H_{T_{\text{ini}}} u_{T_{\text{ini}}}, \quad (45)$$

$$\Leftrightarrow y_{T_{\text{ini}}} = \begin{bmatrix} O_{T_{\text{ini}}} & H_{T_{\text{ini}}} \end{bmatrix} \begin{bmatrix} x_1 \\ u_{T_{\text{ini}}} \end{bmatrix}, \quad (46)$$

Since the recorded sequences (4), (3) are trajectories of system (1) they must satisfy:

$$Y_{T_{\text{ini}}} = \begin{bmatrix} O_{T_{\text{ini}}} & H_{T_{\text{ini}}} \end{bmatrix} \begin{bmatrix} X_1 \\ U_{T_{\text{ini}}} \end{bmatrix}. \quad (47)$$

From (45) we can also uniquely obtain x_1 with $y_{T_{\text{ini}}}$ and $u_{T_{\text{ini}}}$ because Assumption 2 holds. This means there exists a known state-space representation of (1) with $\tilde{x} = \begin{bmatrix} y_{T_{\text{ini}}}^\mathsf{T}, u_{T_{\text{ini}}}^\mathsf{T} \end{bmatrix}^\mathsf{T}$ and some system matrices $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$. Similarly as above, this allows to compute the future trajectory as:

$$y_N = \tilde{O}_N \tilde{x} + \tilde{H}_N u_N, \quad (48)$$

$$\Leftrightarrow y_N = \begin{bmatrix} \tilde{O}_N & \tilde{H}_N \end{bmatrix} \begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ u_N \end{bmatrix} \quad (49)$$

where $\tilde{O}_N$ is the observability matrix and $\tilde{H}_N$ is the Toeplitz matrix of the transformed system $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$. The recorded sequences (4) must satisfy

$$Y_N = \begin{bmatrix} \tilde{O}_N & \tilde{H}_N \end{bmatrix} \begin{bmatrix} Y_{T_{\text{ini}}} \\ U_{T_{\text{ini}}} \\ U_N \end{bmatrix} = \begin{bmatrix} \tilde{O}_N & \tilde{H}_N \end{bmatrix} M. \quad (50)$$

We multiply both sides of the equation with $M^\dagger M$:

$$Y_N M^\dagger M = \begin{bmatrix} \tilde{O}_N & \tilde{H}_N \end{bmatrix} M M^\dagger M. \quad (51)$$

Considering the properties of the Moore-Penrose inverse $(M M^\dagger M = M)$ we can write:

$$Y_N M^\dagger M = \begin{bmatrix} \tilde{O}_N & \tilde{H}_N \end{bmatrix} M, \quad (52)$$

which yields, considering (15),:

$$P^* M = \begin{bmatrix} \tilde{O}_N & \tilde{H}_N \end{bmatrix} M \quad (53)$$

We thus have that the projection of M with P^* is equivalent to the projection of M with the system matrices $\tilde{O}_N$ and $\tilde{H}_N$. From (50), we see that:

$$Y_N = P^* M, \quad (54)$$

which also holds for all linear combinations of the columns of M and Y_N with $\alpha \in \mathbb{R}^T$:

$$Y_N \alpha = P^* M \alpha \qquad \forall \alpha \in \mathbb{R}^T \quad (55)$$

$$\Leftrightarrow y_N = P^* \begin{bmatrix} u_{T_{\text{ini}}} \\ u_N \\ y_{T_{\text{ini}}} \end{bmatrix} \qquad \forall \begin{bmatrix} y_{T_{\text{ini}}} \\ u_{T_{\text{ini}}} \\ u_N \end{bmatrix} \in \text{colspan}(M). \quad (56)$$

We thus have that any $\begin{bmatrix} y_{T_{\text{ini}}}^\mathsf{T}, u_{T_{\text{ini}}}^\mathsf{T}, u_N^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \text{colspan}(M)$ together with the resulting y_N computed via (16) is a sequence of system (1). To show that no other sequences $\begin{bmatrix} y_{T_{\text{ini}}}^\mathsf{T}, u_{T_{\text{ini}}}^\mathsf{T}, u_N^\mathsf{T} \end{bmatrix}^\mathsf{T} \notin \text{colspan}(M)$ are trajectories of system (1), we introduce:

$$P^* = \begin{bmatrix} P_{y_{\text{ini}}}^*, P_{u_{\text{ini}}}^*, P_{u_N}^* \end{bmatrix} \quad (57)$$

228

With (47), (53) and (57), we can write:

$$Y_N = \begin{bmatrix} P^*_{y_{\text{ini}}} O_{T_{\text{ini}}} & P^*_{y_{\text{ini}}} H_{T_{\text{ini}}} + P^*_{u_{\text{ini}}} & P^*_{u_N} \end{bmatrix} \begin{bmatrix} X_1 \\ U_{T_{\text{ini}}} \\ U_N \end{bmatrix}. \quad (58)$$

By Assumption 1 and Assumption 3 the matrix $\begin{bmatrix} X_1^\mathsf{T} & U_{T_{\text{ini}}}^\mathsf{T} & U_N \end{bmatrix}^\mathsf{T}$ has full rank and thus spans the space of all initial states $x_1 \in \mathbb{R}^n$ and sequences $u_{T_{\text{ini}}} \in \mathbb{R}^{T_{\text{ini}}m}$, $u_N \in \mathbb{R}^{Nm}$ for which all possible sequences $y_{T_{\text{ini}}} \in \mathbb{R}^{T_{\text{ini}}p}$ according to (46) and $y_N \in \mathbb{R}^{Np}$ according to (16) are obtained. This shows that $\begin{bmatrix} y_{T_{\text{ini}}}^\mathsf{T}, u_{T_{\text{ini}}}^\mathsf{T}, u_N^\mathsf{T}, \end{bmatrix}^\mathsf{T} \in \text{colspan}(M)$ spans the subspace of all possible sequences of (1) for which (56) and thus (16) holds. ∎

B. Proof of Result 1

We state the Lagrangian for problem (33):

$$\mathcal{L} = \frac{1}{2} g^\mathsf{T} (\lambda_g I) g + \frac{1}{2} y_N^\mathsf{T} \tilde{Q} y_N + \frac{1}{2} v^\mathsf{T} V v \\ - \mu_1^\mathsf{T} (y_N - \tilde{Y}_N g) - \mu_2^\mathsf{T} (\tilde{M} g - b - v) \quad (59)$$

The first order conditions of optimality for problem (33) can be written in terms of the Lagrangian as:

$$\nabla_{y_N} \mathcal{L} = y_N^\mathsf{T} \tilde{Q} - \mu_1^\mathsf{T} = 0 \quad (60)$$

$$\nabla_v \mathcal{L} = v^\mathsf{T} V + \mu_2^\mathsf{T} = 0 \quad (61)$$

$$\nabla_g \mathcal{L} = g^\mathsf{T} (\lambda_g I) + \mu_1^\mathsf{T} \tilde{Y}_N - \mu_2^\mathsf{T} \tilde{M} = 0 \quad (62)$$

$$\nabla_{\mu_1} \mathcal{L} = -y_N^\mathsf{T} + g^\mathsf{T} \tilde{Y}_N^\mathsf{T} = 0 \quad (63)$$

$$\nabla_{\mu_2} \mathcal{L} = -g^\mathsf{T} \tilde{M}^\mathsf{T} + b^\mathsf{T} + v^\mathsf{T} = 0 \quad (64)$$

Inserting (60) and (61) into (62) yields:

$$g^\mathsf{T} (\lambda_g I) + y_N^\mathsf{T} \tilde{Q} \tilde{Y}_N + v^\mathsf{T} V \tilde{M} = 0. \quad (65)$$

Next we insert (63) and (64) in (65) and rearrange, which yields:

$$g = (\lambda_g I + \tilde{M}^\mathsf{T} V \tilde{M} + \tilde{Y}_N^\mathsf{T} \tilde{Q} \tilde{Y}_N)^{-1} \tilde{M}^\mathsf{T} V b. \quad (66)$$

The inverse in (66) always exists because $\tilde{M}^\mathsf{T} V \tilde{M}$ and $\tilde{Y}_N^\mathsf{T} \tilde{Q} \tilde{Y}_N$ are positive semi-definite and $\lambda_g I$ is positive definite. With (64) we obtain:

$$v^* = \tilde{M} (\lambda_g I + \tilde{M}^\mathsf{T} V \tilde{M} + \tilde{Y}_N^\mathsf{T} \tilde{Q} \tilde{Y}_N)^{-1} \tilde{M}^\mathsf{T} V b - b \quad (67)$$

C. Proof of Result 2

Proof: We state the Lagrangian for problem (34):

$$\mathcal{L} = \frac{1}{2} y_N^\mathsf{T} Q y_N + \frac{1}{2} v^\mathsf{T} V v - \mu^\mathsf{T} (y_N - \tilde{P} b - \tilde{P} b) \quad (68)$$

The first order conditions of optimality for problem (34) can be written in terms of the Lagrangian as:

$$\nabla_{y_N} \mathcal{L} = y_N^\mathsf{T} \tilde{Q} - \mu^\mathsf{T} = 0 \quad (69)$$

$$\nabla_v \mathcal{L} = v^\mathsf{T} V + \mu^\mathsf{T} \tilde{P} = 0 \quad (70)$$

$$\nabla_\mu \mathcal{L} = -y_N^\mathsf{T} + v^\mathsf{T} \tilde{P}^\mathsf{T} + b^\mathsf{T} \tilde{P}^\mathsf{T} = 0 \quad (71)$$

Inserting (71) in (69) and the resulting expression in (70) yields:

$$v^* = -(V + \tilde{P}^\mathsf{T} Q \tilde{P})^{-1} \tilde{P}^\mathsf{T} \tilde{Q} \tilde{P} b \quad (72)$$

after rearranging. The inverse in (72) always exists, because $\tilde{P}^\mathsf{T} \tilde{Q} \tilde{P}$ is positive semi-definite and by assumption V is positive definite. ∎

REFERENCES

[1] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, WI, USA: Nob Hill Publishing, 2017.

[2] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1987.

[3] D. F. Delchamps, *State Space and Input-Output Linear Systems*. New York, NY, USA: Springer, 1988.

[4] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, and Q. Zhang, "Nonlinear black-box models in system identification: Mathematical foundations," *Automatica*, vol. 31, no. 12, pp. 1725–1750, Dec. 1995.

[5] L. Ljung, "System Identification," in *Wiley Encyclopedia of Electrical and Electronics Engineering*. Wiley, 2017, pp. 1–19.

[6] W. Favoreel, B. De Moor, and M. Gevers, "SPC: Subspace predictive control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004–4009, 1999.

[7] B. Huang and R. Kadali, *Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-Driven Subspace Approach*. Springer, 2008.

[8] S. Sedghizadeh and S. Beheshti, "Data-driven subspace predictive control: Stability and horizon tuning," *Journal of the Franklin Institute*, vol. 355, no. 15, pp. 7509–7547, 2018.

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning, Second Edition: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[10] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.

[11] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *2019 18th European Control Conference (ECC)*, 2019, pp. 307–312.

[12] J. Berberich, J. Koehler, M. A. Muller, and F. Allgower, "Data-Driven Model Predictive Control with Stability and Robustness Guarantees," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[13] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *arXiv preprint arXiv:2006.01702*, 2020.

[14] D. Alpago, F. Dörfler, and J. Lygeros, "An Extended Kalman Filter for Data-Enabled Predictive Control," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 994–999, 2020.

[15] P. G. Carlet, A. Favato, S. Bolognani, and F. Dörfler, "Data-driven predictive current control for synchronous motor drives," in *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2020, pp. 5148–5154.

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

[17] I. Markovsky, *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2006.

[18] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach*, 1st ed. New York, NY, USA: Cambridge University Press, 2007.

[19] H. V. Henderson and S. R. Searle, "On deriving the inverse of a sum of matrices," *Siam Review*, vol. 23, no. 1, pp. 53–60, 1981.

[20] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, 2018.

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