Subspace predictive control of flexible structures actuated by piezoelectric elements

Van Der Veen, Gijs; Van Wingerden, Jan Willem

DOI
10.1109/CCTA60707.2024.10666613

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the IEEE Conference on Control Technology and Applications, CCTA 2024

2024 IEEE Conference on Control Technology and Applications (CCTA)
August 21-23, 2024. Newcastle upon Tyne, UK

2024 IEEE Conference on Control Technology and Applications (CCTA) | 979-8-3503-7094-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCTA60707.2024.10666613

Subspace predictive control of flexible structures actuated by piezoelectric elements

Gijs van der Veen[1] and Jan-Willem van Wingerden[2]

Abstract— **In this paper we consider data-driven control of vibrations in a flexible structure equipped with piezoelectric transducers. The control algorithm uses elements from the fairly recent data-enabled predictive control framework. In particular, we will develop a real-time implementation of the subspace predictive control algorithm. This algorithm first solves a linear least-squares problem to recursively estimate the observer Markov parameters of the system. With those parameters a predictor is constructed which is used to solve a predictive control problem subject to constraints. The feasibility of the approach is highlighted by applying it to an experimental setup using an efficient implementation. First, this demonstrates that computations can be performed in real-time for a realistic situation. Second, we show how the scheme rapidly adapts when a sudden significant change in structural dynamics is introduced by changing one of the structural parameters.**

I. INTRODUCTION

Vibration control of flexible structures continues to be an area of active research in many engineering disciplines. In particular, there is interest in "smart" structures equipped with deformable materials. Overviews of control design techniques are given in e.g. [1] and [2]. In the control of flexible structures with lightly damped modes, it is advantageous to exploit collocation between actuator and sensor pairs to exploit the fact that this results in transfer functions with a phase that does not drop beyond -180° [3], thereby significantly simplifying and robustifying control design. In practice, however, it has been observed that due to imperfections it is not always possible to achieve perfect collocation. In addition, effects like coupling between different transducer pairs further complicate controller design. Moreover, in many cases of practical interest the characteristics of flexible structures change, e.g., due to variable loading or changing boundary conditions. Hence, adaptive control methodologies for resonant structures have been proposed in [4].

The feasibility of model predictive control (MPC) for vibration control has been demonstrated, e.g. in [5]. In the present work, we aim to combine the benefits of MPC with on-line identification to obtain an adaptive control methodology with a degree of fault-tolerance. Very recently, the field of data-enabled control, in particular the DeePC algorithm [6], [7], received a lot of attention. These ideas are directly connected to the original concepts of subspace predictive control (SPC) [8] presented by Favoreel et al. [9] and Woodley [10]. An improved variant of SPC was presented in [11], [12], based on the predictor-based subspace identification framework [13], [14] and more recently [15] which also introduce a closed-loop DeePC algorithm and showed the equivalence. Compared to the original algorithm, the predictor-based version is inherently asymptotically unbiased in closed-loop operation [16], [15]. The subspace-based algorithm is based on least-squares principles and therefore lends itself to straightforward recursive implementation with attractive numerical properties. In the field of closed-loop SPC, extensions also exist to Hammerstein [17] and LPV [18] model structures, and for repetitive control [19], fault detection [20], estimation [21] and feedforward control [22]. More recently some of these extensions are also made for the DeePC framework [23], [24], [25].

Our contributions are threefold:

- we demonstrate the potential of SPC as an adaptive control methodology in dealing with time-varying system dynamics;
- we present the application of the algorithm to a realistic vibration control example, as opposed to most implementations of SPC thus far, with some challenging computational requirements.
- we carefully highlight the tuning of the inevitable tuning variables.

The remainder of this paper is organized as follows. In Section II the identification framework is introduced. In Section III, the original subspace predictor is presented and the model predictive control problem is formulated. In Section IV, the main contribution of this paper is presented, the experimental evaluation of the proposed data-driven techniques. Finally, conclusions are drawn in Section V.

II. IDENTIFICATION FRAMEWORK

In this section we present the framework for identification and prediction that will be used in the data-driven control scheme. It is assumed throughout that the system operates in a steady operating point for some time, during which its dynamics can be described by a linear, time-invariant model. Variations in the parameters of this model occur slowly. To make this more precise, it is assumed that there is a clear separation between the time constants of the system dynamics and the much slower time constants of parameter variations. With these assumptions, the system, affected by measurement and process noise, admits an innovation state

This work is part of the research programme "Closed-loop wake steering for large densely space wind farms" with project number 17512, which is financed by the Dutch Research Council (NWO).). [1]Netherlands Organisation for Applied Scientific Research (TNO), Optomechatronics dept., Stieltjesweg 1, 2628 CK Delft, The Netherlands. {Gijs.vanderveen}@tno.nl. [2]Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628 CD Delft, The Netherlands. {J.W.vanWingerden}@tudelft.nl.

space representation [26] given by:

$$x_{k+1} = Ax_k + Bu_k + Ke_k, \tag{1a}$$
$$y_k = Cx_k + Du_k + e_k, \tag{1b}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n_u}$, $K \in \mathbb{R}^{n\times n_y}$, $C \in \mathbb{R}^{n_y\times n}$ and $D \in \mathbb{R}^{n_y\times n_u}$. The vectors $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^{n_u}$, $y_k \in \mathbb{R}^{n_y}$ and $e_k \in \mathbb{R}^{n_y}$ are the state vector, input signal, output signal and innovation signal, respectively. The matrix K is the Kalman gain. The innovation sequence e_k is an ergodic zero-mean white noise sequence with covariance matrix $\mathbb{E}\{e_j e_k^\top\} = W\delta_{jk}$, with $W \succ 0$.
The pair (A, C) is observable and the pair $(A, [B\ KW^{\frac{1}{2}}])$ is controllable. This model class is very general in that it encapsulates all LTI MIMO systems with arbitrarily coloured process and measurement noise sequences when those noise sequences have nonsingular rational spectra.

In the representation (1), e_k may be eliminated from the first equation to yield a one-step-ahead predictor:

$$x_{k+1} = \tilde{A}x_k + \tilde{B}u_k + Ky_k, \tag{2a}$$
$$y_k = Cx_k + Du_k + e_k, \tag{2b}$$

where $\tilde{A} \equiv A - KC$ and $\tilde{B} \equiv B - KD$ have been introduced. This predictor forms the basis for the predictor-based subspace identification (PBSID) framework [13].

A. Deriving the data equations

Before deriving the data equations for subspace identification, we will introduce some notation. A stacked sample of input and output data is introduced:

$$z_k = \begin{bmatrix} u_k \\ y_k \end{bmatrix}.$$

The stacked vector $z_k^{(p)}$ is defined as:

$$z_k^{(p)} = \left[\ z_k^\top,\quad z_{k+1}^\top,\quad \cdots,\quad z_{k+p-1}^\top\ \right]^\top,$$

where p denotes the past window size. We also define the (reversed) extended controllability matrix $\mathcal{K}$:

$$\mathcal{K} = \left[\ \tilde{A}^{p-1}[\tilde{B}\ K],\quad \tilde{A}^{p-2}[\tilde{B}\ K],\quad \cdots,\quad [\tilde{B}\ K]\ \right].$$

Note that these matrices contain parameters pertaining to the *predictor* form (2).

Starting from an initial state x_k, the state equation can be propagated forward in time, resulting in:

$$x_{k+p} = \tilde{A}^p x_k + \mathcal{K}z_k^{(p)}. \tag{3}$$

Based on (3) and the output equation, the output at time $k+p$ can then be written as:

$$y_{k+p} = C\tilde{A}^p x_k + C\mathcal{K}z_k^{(p)} + Du_{k+p} + e_{k+p}. \tag{4}$$

Since $\tilde{A}$ has all its eigenvalues inside the unit disc, the term $\tilde{A}^p$ can be made arbitrarily small, i.e. $\|\tilde{A}^p\|_2 \approx 0$, by choosing p sufficiently large. For that reason, the first term on the right hand side of (4) will be neglected.

With N samples available, equation (4) can be repeated to obtain expressions for y_{k+p} to $y_{k+p+N-1}$, resulting in:

$$Y_f = C\mathcal{K}Z_p + DU_f + E_f, \tag{5}$$

with $Y_f = [y_{k+p},\ y_{k+p+1},\ \cdots,\ y_{k+p+N-1}]$, and Z_p, U_f and E_f similarly obtained by horizontal stacking.

1) Relation to the ARX model structure: Taking a closer look at the data equation (4), neglecting the first term, it is seen to have a vector-ARX (VARX) structure. Usually, an ARX model structure prescribes a severely restrictive noise model because it forces the system and noise model to have a common set of poles as seen from the following equation:

$$y_k = \frac{B(z)}{A(z)}u_k + \frac{1}{A(z)}e_k, \tag{6}$$

with z^{-1} the unit backshift operator and:

$$A(z) = 1 - a_1 z^{-1} - \cdots - a_p z^{-p},$$
$$B(z) = b_0 + b_1 z^{-1} + \cdots + b_p z^{-p}.$$

In this context, with sufficiently large p and working with the predictor form, the estimated poles of the VARX model indeed correspond to the poles of the predictor model.
Regarding the ARX model structure (6), the parameters a_i and b_i can explicitly be given as:

$$a_i = C\tilde{A}^{i-1}K, \quad \text{for} \quad i = 1\ldots p, \tag{7a}$$
$$b_i = C\tilde{A}^{i-1}\tilde{B}, \quad \text{for} \quad i = 1\ldots p, \tag{7b}$$
$$b_0 = D. \tag{7c}$$

2) Relation between innovation and predictor representations: In a transfer matrix setting we can represent the system in the following form:

$$y_k = G(z)u_k + H(z)e_k, \tag{8}$$

with:

$$G(z) = C(zI - A)^{-1}B \tag{9}$$
$$H(z) = C(zI - A)^{-1}K + I. \tag{10}$$

A one-step-ahead predictor is then obtained as follows:

$$y_k = G(z)u_k + (H(z) - I)e_k + e_k \tag{11}$$
$$= G(z)u_k + (H(z) - I)H(z)^{-1}(y_k - G(z)u_k) + e_k \tag{12}$$
$$= H(z)^{-1}G(z)u_k + (I - H(z)^{-1})y_k + e_k \tag{13}$$
$$= \tilde{G}(z)u_k + \tilde{H}(z)y_k + e_k. \tag{14}$$

This relation immediately shows the requirements for a stable predictor to exist: the noise model $H(z)$ should possess a stable inverse. Typically, a noise model with non-minimum phase zeros can be represented by a stable spectral factor that results in the same spectral density of the noise process (i.e. with the same second order statistics). Problems arise when the noise model has stable zeros that approach the unit circle. In such cases the predictor approaches marginal stability and the condition that $\|\tilde{A}^p\|_2 \approx 0$ for reasonable p becomes hard to satisfy. The following relations prove the equivalence

between the state-space and input-output representations of the predictor.

$$\begin{aligned}\tilde{G}(z) &= H(z)^{-1}G(z)\\ &= [C(zI-A)^{-1}K+I]^{-1}[C(zI-A)^{-1}B]\\ &= [I-C(zI-\tilde{A})^{-1}K]C(zI-A)^{-1}B\\ &= C(zI-\tilde{A})^{-1}((zI-\tilde{A})(zI-A)^{-1}-KC(zI-A)^{-1})B\\ &= C(zI-\tilde{A})^{-1}B\\ \tilde{H}(z) &= I-H(z)^{-1}\\ &= I-[C(zI-A)^{-1}K+I]^{-1}\\ &= C(zI-\tilde{A})^{-1}K\end{aligned}$$

3) Closed-loop identification issues: Many data-driven control methods ignore or neglect the fact that in a closed-loop control setting the input signal to the system is correlated with the (measurement) noise. An implicit assumption in such methods is that the input signal u_k is uncorrelated with the past noise process e_k. In a closed-loop situation this condition is clearly violated:

$$\mathbb{E}\{u_k e_j^\top\} \neq 0 \quad \text{for} \quad j<k.$$

In the PBSID framework, resulting in data equations of the form in Eq. (5), the estimation is not affected by such correlation issues by segregating the data into collections of "past" and "future" samples. Thus, asymptotically in the number of samples and windows sizes, the only bias in the parameters is due to the finite past window p.

B. Estimating the predictor Markov parameters

Since e_k is a zero-mean white noise innovation sequence, the predictor Markov parameters in (5) can be consistently estimated in a least-squares sense:

$$[\widehat{C\mathcal{K},\ D}] = \arg\min_{[C\mathcal{K},\ D]} \left\| Y_f - [C\mathcal{K},\ D]\begin{bmatrix}Z_p\\ U_f\end{bmatrix}\right\|_F^2, \quad (15)$$

which, for a full-rank data matrix $\left[Z_p^\top,\ U_f^\top\right]^\top$, results in the unique estimate:

$$[\widehat{C\mathcal{K},\ D}] = Y_f\begin{bmatrix}Z_p\\ U_f\end{bmatrix}^+, \quad (16)$$

where $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse. Note that the feedthrough term D, which has been included so far, should only be included when the feedback loop, contains at least a one-sample delay to retain well-posedness of the identification problem.

C. Recursive solution of the parameter estimation problem

In this section the procedure of recursively estimating the parameters in an online setting is briefly discussed. To maintain a compact notation, the least-squares problem (15) is concisely written as:

$$\widehat{\Theta}_k = \arg\min_{\Theta} \|Y_k - \Theta\Phi_k\|_F^2, \quad (17)$$

where the subscript k signifies that data up to time instant k is available and used. The full-rank least-squares solution in (16) is:

$$\widehat{\Theta}_k = Y_k\Phi_k^\top(\Phi_k\Phi_k^\top)^{-1}. \quad (18)$$

When a new sample of input and output data becomes available, the updated solution is given by:

$$\widehat{\Theta}_{k+1} = [Y_k,\ y_{k+1}]\begin{bmatrix}\Phi_k^\top\\ \varphi_{k+1}^\top\end{bmatrix}\left(\Phi_k\Phi_k^\top + \varphi_{k+1}\varphi_{k+1}^\top\right)^{-1}, \quad (19)$$

where y_{k+1} is a new output measurement and φ_{k+1} is a new vector $z_{k+1}^{(p)}$ which is constructed by shifting $z_k^{(p)}$ upwards and appending the new sample z_{k+1}. Then, application of the matrix inversion lemma leads to:

$$\widehat{\Theta}_{k+1} = [Y_k,\ y_{k+1}]\begin{bmatrix}\Phi_k^\top\\ \varphi_{k+1}^\top\end{bmatrix}\left(\Phi_k\Phi_k^\top\right)^{-1} * \left(I - \frac{\varphi_{k+1}\varphi_{k+1}^\top\left(\Phi_k\Phi_k^\top\right)^{-1}}{1+\varphi_{k+1}^\top\left(\Phi_k\Phi_k^\top\right)^{-1}\varphi_{k+1}}\right).$$

In practice a forgetting factor $\lambda < 1$ is introduced to exponentially discount older samples [27]. This ensures that the parameters remain adaptive and is required to maintain a finite covariance matrix:

$$\widehat{\Theta}_{k+1} = [Y_k,\ y_{k+1}]\begin{bmatrix}\Phi_k^\top\\ \varphi_{k+1}^\top\end{bmatrix}\frac{1}{\lambda}\left(\Phi_k\Phi_k^\top\right)^{-1} * \left(I - \frac{\varphi_{k+1}\varphi_{k+1}^\top\left(\Phi_k\Phi_k^\top\right)^{-1}}{\lambda+\varphi_{k+1}^\top\left(\Phi_k\Phi_k^\top\right)^{-1}\varphi_{k+1}}\right).$$

The update equations are written in the following form.

$$P_{k+1} = \frac{1}{\lambda}P_k - \frac{P_k\varphi_{k+1}\varphi_{k+1}^\top P_k}{\lambda+\varphi_{k+1}^\top P_k\varphi_{k+1}}, \quad (20)$$

$$\widehat{\Theta}_{k+1} = \widehat{\Theta}_k + \left(y_{k+1} - \widehat{\Theta}_k\varphi_{k+1}\right)\varphi_{k+1}^\top P_{k+1}. \quad (21)$$

It is finally noted that in propagating the covariance matrix $P_k = (\Phi_k\Phi_k^\top)^{-1}$ in finite precision arithmetic the RLS algorithm is not guaranteed to retain a positive definite covariance matrix. For this reason the simple RLS scheme above is rarely used and instead a square-root algorithm is used [28], [27]. Such algorithms propagate triangular factors of the covariance or information matrix by executing a sequence of orthogonal transformations in each time step. These algorithms are numerically superior to their counterpart described above. The computational complexity of standard RLS as well as the square root counterparts is $\mathcal{O}(p^2(n_u+n_y)^2)$ per iteration, i.e. quadratic in the number of parameters.

Each new data vector appended to the least-squares problem in (17) is in fact just a shifted version of its predecessor with a new sample appended. This knowledge is used in the derivation of so-called fast-array RLS algorithms [27], [29] whose complexity is $\mathcal{O}(p(n_u+n_y))$, thus linear in the number of parameters.

III. Deriving the subspace predictor

Having estimated the predictor Markov parameters from LS problem (15), the next step is to construct an output predictor for a sequence of future outputs. As a starting point (4) is used:

$$y_{k+p} = \widehat{C\mathcal{K}} z_k^{(p)} + \widehat{D} u_{k+p},$$

where the term e_{k+p} has been omitted since for the purpose of prediction we have $\mathbb{E}\{e_{k+p}\} = 0$.
At time instant k, we consider the outputs at time instant $k+1$ up to $k + N_p$, where the arbitrary length of the prediction interval is denoted by $N_p \in [1, \infty)$. Define by $\tilde{y}_k$ a stacked sequence of outputs according to:

$$\tilde{y}_k = \begin{bmatrix} y_{k+1} \\ \vdots \\ y_{k+N_p} \end{bmatrix}.$$

A stacked sequence $\tilde{u}_k$ is defined analogously. Then it is straightforward to show that:

$$\tilde{y}_k = \tilde{\Gamma} z_k^{(p)} + \tilde{H} \tilde{u}_k + \tilde{G} \tilde{y}_k, \tag{22}$$

after defining:

$$\tilde{\Gamma} = \begin{bmatrix} C\tilde{A}^{p-1}[\tilde{B}\ K] & C\tilde{A}^{p-2}[\tilde{B}\ K] & \cdots & C[\tilde{B}\ K] \\ \ddots & \ddots & & \\ & C\tilde{A}^{p-1}[\tilde{B}\ K] & \cdots & C\tilde{A}^{N_p-1}[\tilde{B}\ K] \end{bmatrix},$$

$$\tilde{H} = \begin{bmatrix} 0 & & & & \\ CB & 0 & & & \\ C\tilde{A}B & CB & 0 & & \\ & & \ddots & & \\ C\tilde{A}^{N_p-2}B & & \cdots & CB & 0 \end{bmatrix},$$

$$\tilde{G} = \begin{bmatrix} 0 & & & & \\ CK & 0 & & & \\ C\tilde{A}K & CK & 0 & & \\ & & \ddots & & \\ C\tilde{A}^{N_p-2}K & & \cdots & CK & 0 \end{bmatrix}.$$

These matrices can be constructed directly from the estimated predictor Markov parameters. Note that since the predictor parameters are used, the output prediction $\tilde{y}_k$ appears on both sides of the equation. To obtain the open-loop subspace predictor, the data equation is pre-multiplied with $(I - \tilde{G})^{-1}$ to obtain the open-loop predictor:

$$\tilde{y}_k = \Gamma z_k^{(p)} + H \tilde{u}_k. \tag{23}$$

There are at least two efficient ways to obtain the open-loop output predictor. First, one can simply perform the pre-multiplication with $(I - \tilde{G})^{-1}$, exploiting that the matrix $(I - \tilde{G})$ is lower unit-triangular and allowing efficient forward-substitution to be used. An alternative method, derived in [30], uses a recursive formulation of the forward-substitution procedure to immediately build the predictor matrices, without explicitly forming $(I - \tilde{G})$. This requires less storage, but more looped operations, so the preferred method is implementation-dependent. In both cases it can be shown, by carrying out the pre-multiplication, that the open-loop predictor contains the actual innovation system Markov parameters (A, B, C, D).

A. Setting up the predictive control problem

In this derivation the generalised predictive control (GPC) framework is considered. In the GPC framework, the goal is to, at each time instant, minimise the value of the following objective function:

$$J_k = \sum_{i=1}^{N_p} (y_{k+i} - r_{k+i})^\top Q_y (y_{k+i} - r_{k+i}) + u_{k+i}^\top R_u u_{k+i} + \Delta u_{k+i}^\top R_{\Delta u} \Delta u_{k+i},$$

subject to:

$$\begin{aligned} u_{\min} \leq & u_{k+i} \leq u_{\max}, \qquad i = 1 \dots N_p, \\ \Delta u_{\min} \leq & \Delta u_{k+i} \leq \Delta u_{\max}, \qquad i = 1 \dots N_p. \end{aligned}$$

This objective expresses that the output should track a reference signal r_{k+i}, while minimising the control effort u_{k+i} and control rate Δu_{k+i}[1][2]. Using the notation introduced before and denoting prediction vectors with a tilde, the objective can be reduced to:

$$J_k = (\tilde{y}_k - \tilde{r}_k)^\top \tilde{Q}_y (\tilde{y}_k - \tilde{r}_k) + \tilde{u}_k^\top \tilde{R}_u \tilde{u}_k + \Delta \tilde{u}_k^\top \tilde{R}_{\Delta u} \Delta \tilde{u}_k, \tag{24}$$

after defining block-diagonal matrices $\tilde{Q}_y$, $\tilde{R}_u$ and $\tilde{R}_{\Delta u}$ appropriately.
To obtain the control rate Δu, the following matrices are defined:

$$S_\Delta = \begin{bmatrix} I_m & & & \\ -I_m & I_m & & \\ & \ddots & \ddots & \\ & & -I_m & I_m \end{bmatrix},$$

$$S_0 = \begin{bmatrix} 0 & \cdots & I_m & 0 \\ \vdots & & & \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$

so that:

$$\Delta \tilde{u}_k = S_0 z_k^{(p)} + S_\Delta \tilde{u}_k. \tag{25}$$

The objective function is an inequality constrained quadratic programming (QP) problem in the variables u_{k+i}, $i = 1 \dots N_c$. Here, a basic formulation is shown, into which more sophisticated (in)equality constraints or objective terms can be incorporated while retaining convexity of the optimization problem.

[1] Δu_k can be related to physical rate limits using the approximation $(\Delta u_k)_{\max} \approx \left(\frac{du}{dt}\right)_{\max} \cdot T_s$, with T_s the sample time.

[2] Note that with this formulation perfect steady-state reference tracking can never be achieved if $R_u \neq 0$ and $u_{ss} \neq 0$.

Algorithm 1 Subspace predictive control

Given: $k = 0$, $p > 0$, Φ_0, φ_0, $\tilde{Q}_y$, $\tilde{R}_u$, $\tilde{R}_{\Delta u}$, $N > 0$, $0 < N_c \leq N$

loop

1. Update parameters Θ_k using Eq. 20.
2. Construct the data matrices in Eq. 22 using Θ_k.
3. Solve for the open-loop predictor (Eq. 23).
4. Construct the QP based on (Eq. 24).
5. Solve the QP for $\tilde{u}_k$ using a QP solver.
6. Implement the first element u_{k+1} of the optimal input sequence.
7. $k \leftarrow k + 1$.

end loop

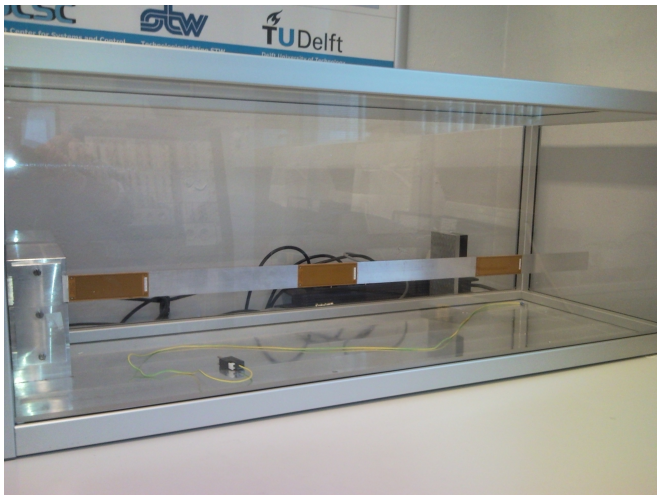

Fig. 1: Photograph of the laboratory setup.

IV. EXPERIMENTAL EVALUATION

The subspace predictive control formulation derived in the previous sections is applied to a vibration control problem on a beam equipped with piezoelectric transducers, see Fig. 1. Two of these transducers are configured as actuators and two as sensors. A schematic top-view of the beam is given in Fig. 2. The beam is an aluminium strip of 95 cm length. At locations 1 and 2, two (almost) collocated sensor-actuator pairs are situated. The actuators are flexible Macro Fiber Composite (MFC) devices, type M8528, from Smart Material Corp. The sensors are connected to high-impedance buffers and the actuators are driven by a high-voltage amplifier. The beam may be used in two different configurations (clamped-free and clamped-pinned), resulting in two very different dynamic behaviours. The beam is an continuum structure with many vibration modes, the lowest of which have very low mechanical damping. The objective of the control algorithm will be to augment damping of the first two vibration modes. During operating, the configuration can switch and the algorithm should track the associated changes in dynamics.

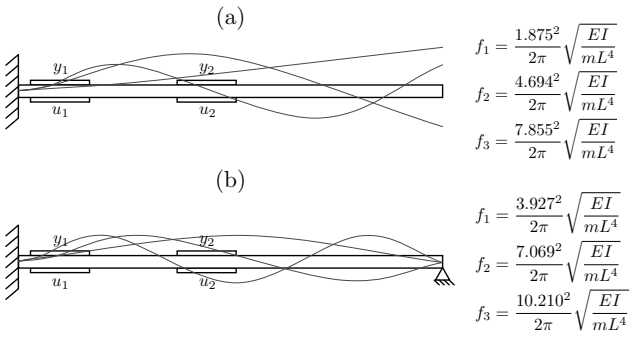

Fig. 2: Schematic view of the two beam configurations and the associated natural vibration modes [31]: (a) clamped, (b) clamped-pinned.

A. Implementation

The algorithm was implemented in C++. For matrix manipulations, use was made of the Eigen template library [32]. The QP is solved using the online active set strategy implemented in the qpOASES package [33], [34]. Although we are dealing with a time-varying QP, the use of an active set strategy is still beneficial since the Hessian is expected to vary slowly most of the time. All code was accessed through Simulink®S-functions and executed using Simulink® Realtime.

To allow the digital control algorithm to operate at a rate of 200 Hz (the maximum achievable on the available PC platform), data was sampled at 2 kHz, then filtered using a second order Butterworth anti-aliasing filter with a corner frequency at 60 Hz and then downsampled to 200 Hz. The control signal was upsampled to 2 kHz and then low-pass filtered at 250 Hz to eliminate the high frequency content introduced by the zero-order hold reconstruction.

B. Parameter selection

Although the presented algorithm requires few parameters to be tuned, those parameters deserve some careful attention. The parameters are:

- The past window dimension p. The value of p determines the number of system parameters that are estimated. In off-line identification, this parameter is usually taken quite large, e.g. several times the maximum expected model order. In the recursive formulation (19), this choice directly affects the computational complexity of the update. Furthermore, estimating a larger number of parameters reduces the convergence rate of the recursive least-squares scheme. In an off-line analysis, a suitable value of p can be obtained, for instance, by validating the hypothesis that the least-squares residual resembles a zero-mean white noise sequence and whether the predictor Markov parameters indeed tend to zero at p. In the current experiment, a value of $p = 25$ was chosen.
- The forgetting factor λ.The value of λ should always be less than 1 to avoid overflow issues when updating

the covariance matrix with new observations. This is also necessary to retain adaptivity of the parameters by forgetting, in an exponential sense, older data. The effective window length is usually expressed as $N \approx 1/(1-\lambda)$ [27]. Depending on the characteristics of new observations, if the effective window is too short, information will be discarded too rapidly and the covariance matrix will become ill-conditioned. Again, in off-line analysis a value for λ can be selected. In this experiment, a value of $\lambda = 0.99995$ was used.

- The prediction horizon N_p is could be chosen such that the prediction interval contains at least one period of each of the modes that are to be damped; its value can be based on a step response of the system. Here, a value of $N_p = 50$ (=0.25 s) was chosen;
- The control horizon N_c determines the number of future inputs that is free. A small N_c reduces the complexity of the QP and further has a smoothing effect on the control signal, which is important for stability. Here, $N_c = 10$ was used;
- The GPC weights, Q_y, R_u and $R_{\Delta u}$, determine the trade-off between control authority (disturbance rejection) and actuator use. In the finite-horizon predictive control problem, these parameters also affect the stability of the closed-loop system. These parameters are usually tuned so as to ensure that the control signals stay within their saturation and rate limits during regular operation.

C. Results

An experiment was performed in which the beam was subjected to random excitations on both actuators. At $t = 10\,\mathrm{s}$ the controller is switched on. Then, at $t = 50\,\mathrm{s}$, an electromagnetic actuator clamps the tip of the beam. This momentarily introduces a high energy perturbation (shock) and results in permanently changed dynamics. During this switching, the controller is kept switched on. From Fig. 3 (the size of the moving window, [10]s, somewhat skews the temporal view of the results) one can see that almost immediately after switching on the controller the second mode is almost completely eliminated. Furthermore, after the change in dynamics, the controller adapts to the new system in seconds, after which the second mode is damped once again. In Fig. 5 we compare the results of identifying the open and closed-loop systems for the first input-output pair. These responses clearly show that the second mode is attenuated by approximately 20 dB in both configurations and the first mode is attenuated by about [10]dB.

V. CONCLUSIONS

In this paper we have successfully demonstrated how subspace predictive control can be applied to vibration control problems in flexible structures. Key advantages are the adaptation to changes in dynamics and the limited set of tuning parameters. Although the scheme may be computationally complex, it can easily be run on present-day processors by exploiting structure in the algorithm. To maintain proper conditioning of the covariance matrix in a forgetting RLS scheme it is necessary that conventional persistence of excitation conditions are satisfied. This can be achieved by applying an appropriate reference perturbation that is as small as possible (to avoid loss of performance). In future work we intend to address these issues within the operational requirements of real systems. Stability and robustness of SPC under general conditions are also still largely open problems [35].

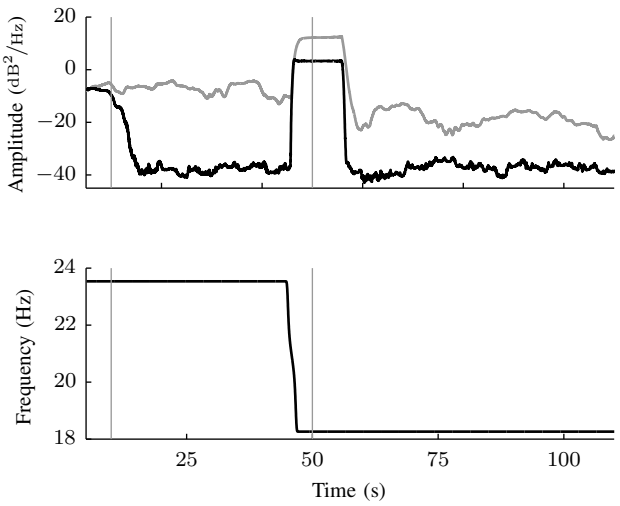

Fig. 3: Plot showing the spectral amplitude and frequency of the second resonance measured at sensor 1 (y_1) as a function of time: (gray) open-loop, (black) closed-loop

REFERENCES

[1] A. Preumont and K. Seto, *Active Control of Structures*. Wiley, 2008.

[2] S. O. R. Moheimani and A. J. Fleming, *Piezoelectric Transducers for Vibration Control and Damping*. Springer, 2006.

[3] A. Preumont, *Vibration control of active structures, an introduction*. Kluwer Academic Publishers, 2002.

[4] D. Niederberger, A. Fleming, S. O. R. Moheimani, and M. Morari, "Adaptive multi-mode resonant piezoelectric shunt damping," *Smart Materials and Structures*, vol. 13, no. 5, p. 1025, 2004.

[5] A. Wills, D. Bates, A. Fleming, B. Ninness, and S. Moheimani, "Model predictive control applied to constraint handling in active noise and vibration control," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 1, pp. 3 –12, 1 2008.

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

[7] J. Coulson, J. Lygeros, and F. Dorfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, 2021.

[8] 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)*, 2022, pp. 2111–2116.

[9] W. Favoreel, B. De Moor, P. Van Overschee, and M. Gevers, "Model-free subspace-based LQG-design," in *Proc. of the American Control Conference*, vol. 5, 1999, pp. 3372 –3376.

[10] B. R. Woodley, "Model free subspace based H_∞ control," Ph.D. dissertation, Stanford University, 2001.

[11] J. Dong, M. Verhaegen, and E. Holweg, "Closed-loop subspace predictive control for fault tolerant MPC design," in *Proceedings of the 17th IFAC World Congress*, vol. 17, 2008.

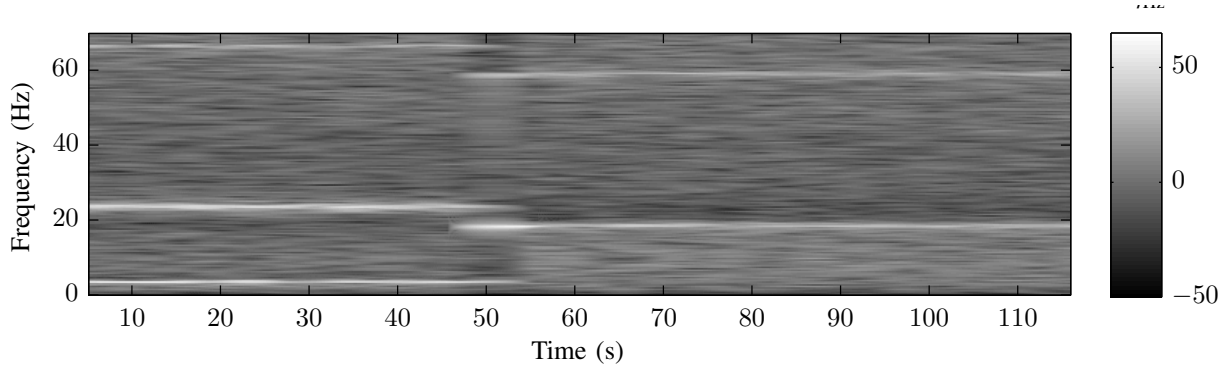

Fig. 4: Waterfall plot of the output spectrum of sensor 1 (y_1) as a function of time.

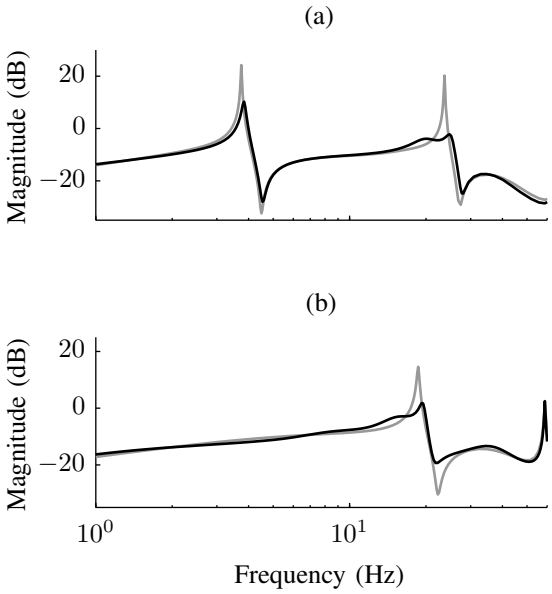

Fig. 5: Identified frequency responses of the open-loop (gray) and closed-loop (black) system from disturbance u_1 to output y_1 for configurations (a) and (b).

[12] R. Hallouzi, "Multiple-model based diagnosis for adaptive fault-tolerant control," Ph.D. dissertation, Delft University of Technology, 2008.

[13] A. Chiuso, "On the relation between CCA and predictor based subspace identification," *IEEE Transactions on Automatic Control*, vol. 52, no. 10, pp. 1795–1812, 2007.

[14] G. van der Veen, J. W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen, "Closed-loop subspace identification methods: an overview," *IET Control Theory & Applications*, vol. 7, no. 10, pp. 1339–1358, 2013.

[15] R. Dinkla, S. P. Mulders, T. Oomen, and J. W. van Wingerden, "Closed-loop data-enabled predictive control and its equivalence with closed-loop subspace predictive control," *arXiv preprint arXiv:2402.14374*, 2024.

[16] R. Dinkla, S. P. Mulders, J. W. van Wingerden, and T. Oomen, "Closed-loop aspects of data-enabled predictive control," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1388–1393, 2023, 22nd IFAC World Congress.

[17] B. Kulcsár, J. W. van Wingerden, J. Dong, and M. Verhaegen, "Closed-loop subspace predictive control for hammerstein systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 2604–2609.

[18] J. Dong, B. Kulcsár, J. W. van Wingerden, and M. Verhaegen, "Closed-loop subspace predictive control for linear parameter varying systems (i)-the nominal case," in *2009 European Control Conference (ECC)*. IEEE, 2009.

[19] S. T. Navalkar, J. W. van Wingerden, E. van Solingen, T. Oomen, E. Pasterkamp, and G. Van Kuik, "Subspace predictive repetitive control to mitigate periodic loads on large scale wind turbines," *Mechatronics*, vol. 24, no. 8, pp. 916–925, 2014.

[20] Y. Liu, J. Frederik, R. M. Ferrari, P. Wu, S. Li, and J. W. van Wingerden, "Fault-tolerant individual pitch control of floating offshore wind turbines via subspace predictive repetitive control," *Wind Energy*, vol. 24, no. 9, pp. 1045–1065, 2021.

[21] Y. Liu, A. K. Pamososuryo, R. Ferrari, T. G. Hovgaard, and J. W. van Wingerden, "Blade effective wind speed estimation: A subspace predictive repetitive estimator approach," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 1205–1210.

[22] R. Dinkla, T. Oomen, J. W. van Wingerden, and S. P. Mulders, "Data-driven lidar feedforward predictive wind turbine control," in *2023 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2023, pp. 559–565.

[23] T. B. Hamdan, P. Coirault, G. Mercère, and T. Dairay, "Data enabled predictive control of LPV systems," *Control Engineering Practice*, vol. 149, p. 105969, 2024.

[24] C. Verhoek, H. S. Abbas, R. Tóth, and S. Haesaert, "Data-driven predictive control for linear parameter-varying systems," *IFAC-PapersOnLine*, vol. 54, no. 8, pp. 101–108, 2021.

[25] K. Zhang, R. Zuliani, E. C. Balta, and J. Lygeros, "Data-enabled predictive repetitive control," *arXiv preprint arXiv:2403.11883*, 2024.

[26] M. Verhaegen and V. Verdult, *Filtering and System Identification; A Least Squares Approach*, 1st ed. Cambridge, UK: Cambridge University Press, 2007.

[27] A. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[28] M. Verhaegen, "Round-off error propagation in four generally-applicable, recursive, least-squares estimation schemes," *Automatica*, vol. 25, no. 3, pp. 437 – 444, 1989.

[29] I. Houtzager, J. W. van Wingerden, and M. Verhaegen, "Fast-array recursive closed-loop subspace model identification," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 96–101, 2009, 15th IFAC Symposium on System Identification.

[30] J. Dong, "Data driven fault tolerant control: a subspace approach," Ph.D. dissertation, Delft University of Technology, 2009.

[31] D. J. Inman, *Engineering Vibration*, 2nd ed. ed. Prentice-Hall, Inc., 2001.

[32] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

[33] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[34] H. Ferreau, A. Potschka, and C. Kirches, "qpOASES webpage," http://www.qpOASES.org/, 2007–2014.

[35] J. Berberich and F. Allgöwer, "An overview of systems-theoretic guarantees in data-driven model predictive control," *arXiv preprint arXiv:2406.04130*, 2024.