

Data-Driven Predictive Control With Online Adaption: Application to a Fuel Cell System

Lukas Schmitt[✉], Julius Beerwerth, Matthias Bahr, and Dirk Abel

Abstract—Fuel cell systems constitute an electrochemical energy conversion system increasingly used in stationary and mobile applications. Complying with operational limits in transient operation can be achieved by model-based predictive control algorithms. The key challenge arises from the identification of suitable models for embedded real-time optimization. This article presents a data-driven predictive control approach for the air path and power control of a fuel cell system. In particular, we use data-enabled predictive control (DeePC) based on a concise system representation using column subset selection (CSS). The impact of problem formulation, regularization, and different solvers for quadratic programs (QPs) on the turnaround time on embedded hardware is investigated. In addition, we provide an online update algorithm for the system representation to account for the operating regions not contained in the initial dataset. The proposed approach is validated on a high-fidelity fuel cell system simulation and hardware-in-the-loop (HiL) experiments. We demonstrate safe and fast closed-loop control using the column subset algorithms for a comprehensive dataset and reduction in closed-loop cost for unknown operating areas of up to 25%. The control algorithm and the update algorithm are shown to be real-time feasible on a single-core embedded hardware.

Index Terms—Data-driven optimal control, fuel cell systems, optimal operation and control of power systems, predictive control, real-time optimal control.

I. INTRODUCTION

PROTON-EXCHANGE membrane fuel cells convert chemical energy directly into electrical energy. In contrast to thermal processes, such as the combustion of hydrogen or fossil fuels, the efficiency is not limited by the Carnot efficiency, but in terms of Gibbs' potential [1]. With the overall reaction $2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$, only water and no greenhouse gases or pollutants are emitted in the energy conversion process. A prerequisite for reliable operation and longevity of the fuel cell is an appropriate supply of oxygen and hydrogen in tight stoichiometric limits. To this end, supporting components

Manuscript received 26 January 2023; revised 29 January 2023 and 30 June 2023; accepted 4 July 2023. Date of publication 4 August 2023; date of current version 29 December 2023. This work was supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) through the AiF (German Federation of Industrial Research Associations eV) based on a decision taken by the German Bundestag. Recommended by Associate Editor A. Mesbah. (*Corresponding author: Lukas Schmitt.*)

Lukas Schmitt and Dirk Abel are with the Institute of Automatic Control, RWTH Aachen University, 52074 Aachen, Germany (e-mail: l.schmitt@irt.rwth-aachen.de; d.abel@irt.rwth-aachen.de).

Julius Beerwerth was with the Institute of Automatic Control, RWTH Aachen University, 52074 Aachen, Germany. He is now with the Chair of Embedded Software, RWTH Aachen University, 52074 Aachen, Germany (e-mail: beerwerth@embedded.rwth-aachen.de).

Matthias Bahr is with The hydrogen and fuel cell center ZBT GmbH, 47057 Duisburg, Germany (e-mail: m.bahr@zbt.de).

Digital Object Identifier 10.1109/TCST.2023.3293790

must be integrated into the fuel cell system. An electric compressor provides adequate air supply, while throttles can be used to adjust the pressure inside the stack. The anode system regulates the hydrogen supply from a pressurized tank system using throttles. Excess hydrogen is recirculated into the anode using a recirculation blower or intermittently purged. Excess heat must be dissipated by a (typically water-based) cooling system to avoid damaging the fuel cell stack. Power electronics regulate the stack current to meet the desired net power output.

The compressor limits the transient response of the cathode system. It must therefore be operated at the boundary of its operating region while safely avoiding aerodynamic instabilities, i.e., compressor surge. Moreover, the sensitivity of the pressure dynamics and air mass flow in the cathode is high with respect to the throttle actuation, especially for a nearly closed throttle position. Stoichiometry inside the stack, however, depends on the mass flow through the cathode such that precise throttle actuation is necessary for reliable fuel cell operation. Overall, due to the nonlinear, multi-input multi-output system characteristics and constraints on inputs and outputs, fuel cell system control is a challenging problem.

After seminal work of the model-based feedforward and feedback control by Pukrushpan et al. [2], the explicit considerations of constraints in the problem formulation initiated research on the model-based predictive control algorithms for fuel cell control. Those methods often require a system identification step, followed by controller synthesis. The most pervasive class based on this paradigm is model predictive control (MPC) [3]. To limit the computational effort needed for the solution of the underlying optimization problem, Acre et al. [4], Danzer et al. [5], Vrlić [6], and Nebeluk and M. Ławryńczuk [7] use linearized models, of which Acre et al. [4] demonstrate real-time explicit linear MPC. However, closed-loop constraint satisfaction in practical application significantly depends on the model accuracy. The restrictive class of linear models can impair prediction quality and, in turn, closed-loop performance and constraint satisfaction significantly. Nonlinear MPC has been shown to provide good closed-loop control over a wide range of operating points [8], [9]. Although advantages of predictive control have been shown for transient operation of fuel cell systems, important challenges remain. The closed-loop real-time solution of the nonlinear, nonconvex optimization problem is challenging on embedded hardware. Modeling and system identification for control and embedded optimization in predictive control is subject to research [10], [11]. Different

state-space models have been proposed for fuel cell systems (see [9], [12], [13], [14]).

In contrast, data-driven control avoids the model identification step but directly relies on input–output data. Subspace predictive control (SPC) is an early example of this class of control methods [15], [16], [17], [18]. Closely related, in fact, equivalent for an underlying linear time-invariant (LTI) system [19], is the recently proposed data-enabled predictive control (DeePC) framework [20], [21], [22], [23], [24] based on the behavioral system theory [25], [26]. Analogously to SPC and MPC, an optimization problem must be solved in every time step to calculate the optimal control input. The system representation, however, is formulated using prerecorded input–output trajectories collected in a Hankel-structured data matrix and the most recent input–output data. For controllable LTI systems, the fundamental lemma provides constructive guarantees for a behavioral system representation based on a persistently exciting input and the corresponding output signal [27].

However, the representativeness of the input–output signal is not given for noisy data or nonlinear systems, where the superposition principle for trajectories similar to motion primitives does not hold in general. Due to several successful applications of DeePC to nonlinear systems (see [28], [29], [30], [31], [32]), research effort has been dedicated to develop a theoretic foundation for nonlinear systems and noisy data and improve closed-loop performance. First, the regularization introduced already in [21] has been rigorously derived based on the assumptions on the noise generation process and related to an implicit system identification [22], [23], [24]. Second, open-loop [23], [33] and closed-loop guarantees of DeePC such as practical exponential stability [34] and robust constraint satisfaction [35] have been developed for noisy LTI systems. An extension to robust stability using terminal ingredients for these classes of system is presented in [36] and [37]. Most recently, practical stability guarantees for nonlinear systems have been set forth [38]. Third, online optimization has been identified as a bottleneck. Therefore, structure exploiting optimization has been proposed to reduce the computational complexity of the underlying quadratic program (QP) [39]. Fourth, online adaptation of the data matrix is crucial for proving practical closed-loop stability based on exploiting local linear approximations of the underlying system [38]. In addition, an adaptation procedure for the behavioral system representation might be necessary for the underlying nonlinear systems, especially when the original data matrix does not cover all the operating regions. For the nonparametric system model of DeePC, past data are directly used as a system description such that these data must be updated accordingly. In [32], [39], and [40], the data matrix is always updated with the most recent data, which can impair closed-loop control performance, e.g., in steady-state operation.

Much of the aforementioned work must be combined for a real-world applicable DeePC algorithm. Especially the online solution of the underlying optimization problem on embedded hardware can be challenging. A concise system representation consisting of as few trajectories as possible facilitates the application of DeePC on hardware with limited resources

for real-world problems. In this work, we illustrate the use of DeePC for the real-time control of a fuel cell system simulation. Using a behavioral system representation, we avoid the model structure selection and identification steps. Instead, the identification step is implicitly solved with the optimal control problem based on data representation in every time step based on the most recent input–output data. We extend our work from [41] in the analysis of embedded optimization for different regularization schemes. Furthermore, we extend the DeePC algorithm with an online adaptation algorithm for the data-based system representation. The improved closed-loop control performance in the operating regimes that are not included in the initial dataset is shown through a high-fidelity fuel cell system simulation study. Our contributions are summarized in the following.

- 1) Column subset selection (CSS) algorithms can be used to formulate a concise behavioral model representation.
- 2) Real-time feasibility on embedded hardware is demonstrated for a sparse and a dense problem formulation.
- 3) Online adaption of the data matrix extends the applicability of DeePC in practical applications. Hardware-in-the-loop (HiL) experiments confirm real-time feasibility on single-core embedded hardware.

This article is organized as follows. In Section II, we present our fuel cell system application and the control objective. Section III details the rationale of our control concept and we state the DeePC optimization problem and our proposed online adaptation algorithm. Closed-loop control results and HiL experiments are presented in Section IV. We conclude with Section V.

II. PROBLEM DESCRIPTION

A. Fuel Cell System Simulation Model

The overall fuel cell system model [42] consists of a compressor model, the anode and cathode media circuit, the cooling circuit, and a detailed 2-D–1-D fuel cell stack model of a proton-exchange membrane stack [43]. It contains more than 110 system states, displays nonlinear behavior, is largely based on physical relations, and considers all the relevant phenomena in the millisecond range and slower. The simulation model is developed in AVL CruiseM/Model.CONNECT and exported via functional mock-up unit (FMU) to enable the cosimulation in SIMULINK with the control algorithm. A schematic of the overall system is shown in Fig. 1.

The core of the overall system model is the real-time capable and locally resolved fuel cell model developed in MATLAB/SIMULINK. This segmented 2-D–1-D fuel cell stack model is based on the model published in [43]. It has been extended with dynamic components and influences to include the effects of highly dynamic processes on the performance of the fuel cell [42]. Due to the multidimensional resolution, the electrochemical and control-relevant processes within the fuel cell can be considered locally and in more detail at low computational costs. Electrochemical processes such as the internal water transport between the anode and cathode and local effects due to internal humidification and reaction

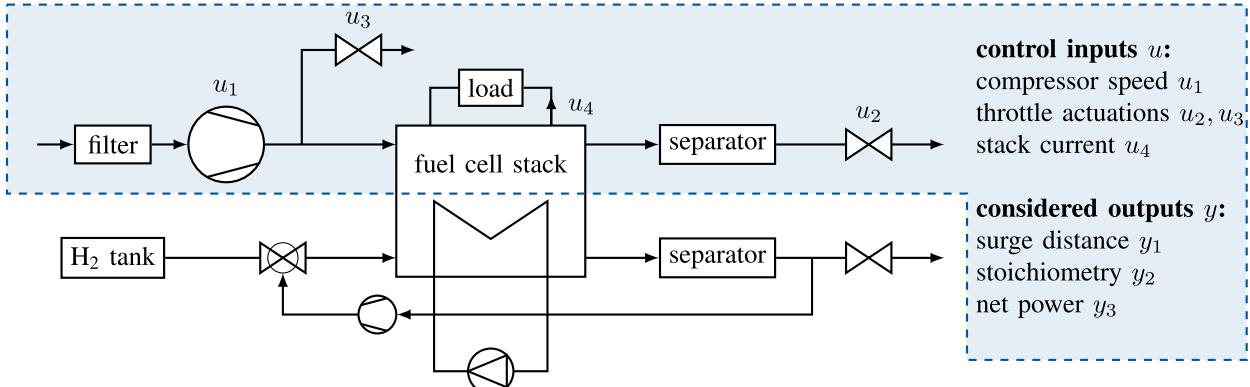


Fig. 1. Fuel cell system with distinct cathode, anode, cooling, and electric system based on [1] and adapted to our system architecture. Our control algorithm regulates the cathode and electric system in the blue-shaded part of the fuel cell system. Depending on the net power output reference, a specific mass flow of ambient air containing the necessary oxygen must be fed to the stack while regulating the current drawn from the stack.

media availability are considered. The fuel cell stack model is calibrated and validated on measurements.

The system model developed in AVL CruiseM and AVL Model.CONNECT includes, in addition to the mentioned stack model, the relevant media circuits and their balance of plant (BoP) components. Since the presented control algorithm refers to the cathode supply and power control, see Section II-B, only the affected components of the cathode are described hereafter. Further elaborations regarding the remaining components of the system can be found in [42]. The most important BoP component in the cathode path is the compressor. It ensures adequate air supply for the fuel cell stack and directly affects the stoichiometry. Due to its high power consumption, the compressor is the major influence on the efficiency of the fuel cell system. To accurately capture this influence, a detailed model for the CT-17-700.GB compressor by the Celeroton AG based on experimental data is formulated.

Furthermore, models of the bypass throttle and the outlet pressure valve are implemented. The bypass valve is controlled by its opening angle and influences the proportion of the air mass flow that passes through the fuel cell stack. Especially for partial load demand, the minimum air mass flow of the compressor can be partially bypassed at the stack to prevent the stack from drying out and to control the cathode stoichiometry. The outlet pressure of the cathode is controlled by the opening angle of the outlet pressure valve. Thus, the pressure and the corresponding control inputs are physically strongly coupled with the operating point of the cathode compressor and the bypass valve actuation. The resulting pressure ratio over the compressor is essential for operating in the feasible region of the compressor map. Water separators are installed after the stack on the cathode and anode path.

B. Control Objective

Given the fuel cell system, the control task is to track an externally provided net power reference

$$P_{\text{net}} = P_{\text{fc}} - P_{\text{cp}} \quad (1)$$

where P_{fc} is the fuel cell power output and P_{cp} the parasitic compressor power. For longevity and durability, the control

algorithm must additionally guarantee adherence to the operational constraints

$$\lambda_{\min} \leq \lambda \leq \lambda_{\max} \quad (2)$$

$$s_{d,\min} \leq s_d \leq s_{d,\max}. \quad (3)$$

The compressor must be operated at a certain surge distance $s_d \in [2(g/s), 8(g/s)]$ from the surge line. The surge distance describes the excess mass flow for a given pressure ratio over the compressor to avoid compressor surge and keep the compressor operative. It is defined as the horizontal distance to the surge line in the compressor map. The stoichiometry of hydrogen and oxygen in the stack must always remain within predefined limits $\lambda \in [1.75, 3]$. Since no external humidification is provided, the stoichiometric limit allows for sufficient self-humidification and prevents the fuel cell stack from drying out. From the control objective and the constraints, we compose the output vector $y = [y_1, y_2, y_3]^{\top} = [s_d, \lambda, P_{\text{net}}]^{\top}$.

We use the closed-loop cost

$$c = \sum_{k=0}^{n_s} (\|y_k - r_k\|_Q + \|\Delta u_k\|_R) \quad (4)$$

where $Q \in \mathbb{R}^{p \times p}$ is the weighting matrix on the tracking error and $R \in \mathbb{R}^{m \times m}$ penalizes the change in control inputs, based on normalized inputs and outputs for evaluating the closed-loop control quality for a test cycle with n_s time steps. The normalized closed-loop tracking cost is denoted by \bar{c} . We consider an experiment valid if the culminated number of time steps with constraint violations is $n_v \leq 10$ and the constraint violations are below a certain threshold. In real-world applications with unknown model-plant mismatch, exact constraint satisfaction cannot be guaranteed without further assumptions. Constraint tightening can be used to safeguard against inaccurate modeling.

III. CONTROL CONCEPT

This section describes the rationale of our control strategy. We introduce the behavioral system representation, the resulting optimal control problem, and the numerical solution.

A. Preliminaries

For the control theoretic preliminaries, we assume an LTI system. Given a sequence of inputs $u = \{u_k\}_{k=1}^T$ of length T , where $u_k \in \mathbb{R}^m$, we define the Hankel matrix with depth L as

$$H_L(u) = \begin{bmatrix} u_1 & u_2 & \dots & u_{T-L+1} \\ u_2 & u_3 & \dots & u_{T-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L+1} & \dots & u_T \end{bmatrix}. \quad (5)$$

Definition 1 (Persistency of Excitation [27]): A sequence of signals u is called persistently exciting of order L if the Hankel matrix $H_L(u)$ has full row rank, i.e., $\text{rank}(H_L(u)) = mL$.

For the matrix $H_L(u)$ to have full row rank, the length of the initial trajectory is lower bounded by $T \geq (m+1)L-1$ [25]. The so-called fundamental lemma [27] connects controllability, persistence of excitation, and the column span of the Hankel matrix to formulate a behavioral representation of LTI systems:

Theorem 1 [44]: Suppose $\{u_k, y_k\}_{k=1}^T$ is a trajectory of a controllable LTI system G , where u is persistently exciting of order $L+n$ and n is the system order. Then, $\{\bar{u}_k, \bar{y}_k\}_{k=1}^L$ is a trajectory of G if and only if there exists $g \in \mathbb{R}^{T-L+1}$ such that

$$\begin{bmatrix} H_L(u) \\ H_L(y) \end{bmatrix} g = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix}. \quad (6)$$

According to this theorem, all the trajectories of an unknown LTI system can be constructed from a single, persistently exciting trajectory. Equivalently, the subspace spanned by the columns of the Hankel matrix is equal to the subspace of possible trajectories of the underlying LTI system. We define

$$U_L = [u^1 \ u^2 \ \dots \ u^T] \quad (7)$$

where $u^i = [u_k^i; u_{k+1}^i; \dots; u_{k+L-1}^i]$ denotes a vector obtained by vertically concatenating the input vectors of a sequence of length L . Likewise, the corresponding output vectors are concatenated into

$$Y_L = [y^1 \ y^2 \ \dots \ y^T]. \quad (8)$$

It has been found that the originally used Hankel structure is not required, but rather any columnwise trajectory matrix which has full row rank, i.e., $\text{rank}(U_L) = mL$ can be used in (6) instead of the Hankel matrices, if some additional, technical conditions are satisfied [19], [22].

SPC originally proposed in [15] and [16] elegantly combines subspace identification and predictive control [17]. Based on the system representation (7) and (8), a linear transformation from past input–output data and future input data to the predicted future output trajectory is calculated. Corresponding to the trajectory length of $L = T_{\text{ini}} + N$, the matrices are divided into past data of length T_{ini} and future data of length N , denoted by subscripts p and f , respectively,

$$U_L = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \quad Y_L = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix} \quad (9)$$

i.e., $U_p \in \mathbb{R}^{mT_{\text{ini}} \times (T-L+1)}$, $Y_p \in \mathbb{R}^{pT_{\text{ini}} \times (T-L+1)}$, $Y_f \in \mathbb{R}^{pN \times (T-L+1)}$, and $U_f \in \mathbb{R}^{mN \times (T-L+1)}$. The subspace identification procedure constructs a matrix $K \in \mathbb{R}^{pn \times ((m+p)T_{\text{ini}} + mN)}$ mapping from past data and future inputs to future outputs analogously to [19]

$$Y_f = K \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}. \quad (10)$$

The optimal linear, multistep autoregressive exogenous (ARX) predictor K^* can be calculated by means of the least-squares optimization problem

$$K^* = \arg \min_K \|K \underbrace{\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}}_M - Y_f\|_F^2 \quad (11)$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and uniqueness holds under full rank conditions [18], [25].

When facing real-world problems, the essential assumption of an underlying deterministic LTI system is often invalid. As a consequence, persistence of excitation in the sense of generating an output sequence that is representative of the system's behavior cannot be guaranteed. Even with a rich initial dataset and a Hankel matrix with full row rank, ensuring an adequate system representation through data trajectories is a major challenge for nonlinear systems [28]. In addition, a data matrix built from noisy data or nonlinear systems will most likely have full rank without revealing the subspace of the system dynamics.

B. System Representation and Data Selection

The system representation in (6) can be formulated using (7) and (8) for the most recent inputs u_{ini} and outputs y_{ini} as

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}. \quad (12)$$

where u and y are the next N future inputs and outputs, respectively. Given the past sequences u_{ini} , y_{ini} , and the future input sequence u , the first three blocks can be used to calculate g . The output prediction results from the last block as $Y_f g = y$ [45].

In practice, two approaches have been proposed to extend the applicability from LTI systems to general nonlinear systems: First, collecting enough data with sufficiently large T_{ini} and significantly more columns than theoretically necessary for LTI systems. Second, regularization is implemented to single out a combination of possibly few columns in each time step. When implementing the algorithm in resource-constrained embedded hardware, however, the system representation must be compact to reduce the impact on memory and optimization runtime. Therefore, it is essential to capture the system behavior in as few columns as possible.

For practical application, we propose the use of CSS algorithms for a low-rank approximation $M_c \in \mathbb{R}^{(mL+pT_{\text{ini}}) \times n_c}$ of a data matrix $M \in \mathbb{R}^{(mL+pT_{\text{ini}}) \times (T-L+1)}$ with a limited number

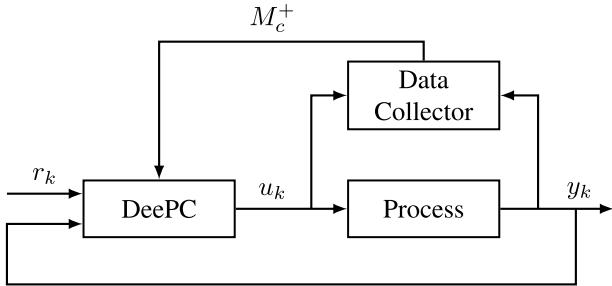


Fig. 2. Schematic visualization of the online update for the data matrix. The data matrix is part of the equality constraints in the underlying QP.

of columns, i.e., $n_c \ll (T - L + 1)$. Since n_c is larger than the row rank of the data matrix, the reconstruction error $\|M - M_c M_c^\dagger M\|_F$, where M_c^\dagger denotes the Moore–Penrose pseudoinverse of M_c , can then not be used anymore to evaluate the CSS. In particular, we use the deterministic variants of norm sampling, iterative norm sampling, leverage score sampling, sampling based on the strong rank-revealing QR (sRRQR) factorization, and, for reference, random sampling for the normalized Hankel data matrix of an input sequence U_L . With norm sampling, we select the n_c columns of the data matrix with the highest squared two-norm. For iterative norm sampling, after each column selection, the projection of the data matrix onto the space spanned by the already selected columns is subtracted from the data matrix before again selecting the column with the largest norm. Leverage score sampling selects columns based on the squared two-norm of the top- k truncated right singular vectors, which is a measure of how important a column is in composing the column space. Finally, we apply the sRRQR decomposition, which generates a specific QR decomposition based on column pivoting such that the numerical rank of the data matrix with selected columns is improved. For iterative norm sampling and sRRQR-based sampling, the row rank limits the number of columns that can be selected. We alter those algorithms to repeatedly select at most mL columns, delete these columns from U_L , and repeat the algorithm until n_c columns are selected. For more detailed information, we refer to [41] and [46].

C. Measure for the Quality of a Column Subset

To find a measure for evaluating the quality of the selected columns, a concept from the field of coresets construction for regression problems described in [47] is adapted. Given the data matrix M and the target matrix Y_f , the optimal solution to the original regression problem is defined by (11). A coresset of size $n_c < T - L + 1$ is a subset M_c of the columns of M and the corresponding elements $Y_{f,c} \in \mathbb{R}^{pn \times n_c}$ of Y_f . Then, the solution to the coresset regression problem is defined as

$$K_c^* = \arg \min_K \|KM_c - Y_{f,c}\|_F^2. \quad (13)$$

To evaluate the quality of the coresset, the least-squares error of K on the original regression problem

$$\mathcal{E}(K) = \|KM - Y_f\|_F^2 \quad (14)$$

is evaluated for both the solution of the full regression problem K^* and the solution of the coresset regression problem K_c^* . The

Algorithm 1 Online Adaption for DeePC Data Matrix

Input: data matrix M_c , buffered input-output data u_b, y_b

Form $H_L(u_b) = \begin{bmatrix} U_{p,b} \\ U_{f,b} \end{bmatrix}$, $H_L(y_b) = \begin{bmatrix} Y_{p,b} \\ Y_{f,b} \end{bmatrix}$, $M_b = \begin{bmatrix} U_{p,b} \\ Y_{p,b} \\ U_{f,b} \end{bmatrix}$

Concatenate $M = [M_c, M_b]$

Calculate K_c^* from (11) using M_c

Calculate K^* from (13) using M

Evaluate subset prediction quality ε with (15)

if $\varepsilon > \varepsilon_{\text{tol}}$ **then**

$M_c^+ = \text{css}(M_c)$ ▷ Apply CSS algorithm

else

$M_c^+ = M_c$

end if

Output: updated data matrix M_c^+

ratio of the two errors defined as

$$\varepsilon = \frac{\mathcal{E}(K_c^*)}{\mathcal{E}(K^*)} \quad (15)$$

is used to evaluate the quality of a column subset. The presented measure acts as a surrogate for the reconstruction error when sampling more than $\text{rank}(M)$ columns of a matrix M and will be of practical use when discussing the online update strategies for the data trajectories.

D. Online Adaption Algorithm

The online-generated data can be used directly to update columns of the data matrix (see Fig. 2). To this end, the data collector buffers the input–output data (denoted by subscript b), triggers an update algorithm for the data matrix, and feeds back the updated data matrix M_c^+ to the DeePC algorithm. In particular, past input and output data trajectories are normalized and stored in a buffer, which is fed to the adaptation algorithm every T_a seconds. First, the update algorithm generates Hankel-structured matrices $H_L(u_b)$ and $H_L(y_b)$ for the buffered inputs and outputs. The new data matrix $M_b \in \mathbb{R}^{(m+p)T_{\text{ini}}+mN \times n_a}$, where $n_a = (T_a/T_s - L + 1)$, is horizontally concatenated with the current data matrix M_c to form the extended data matrix M . Subsequently, the optimal predictors for the current and extended data matrix are calculated and the relative error (15) is evaluated. Above the threshold ε_{tol} , the data matrix is updated by applying a CSS algorithm to the extended data matrix to select exactly n_c columns. Otherwise, the current data matrix is not altered and fed back to the DeePC algorithm. The update procedure is summarized in Algorithm 1.

For updating the data trajectories online, the offline calculated data matrix is divided into a set of fixed columns and a set of free columns, where the updating algorithm operates. With that partitioning, it is ensured that a solid base of trajectories is always present.

E. Data-Enabled Predictive Control

DeePC is a relatively recently proposed predictive control algorithm based purely on prerecorded input–output data trajectories [21], [27]. For the LTI systems, it was proven to be

equivalent to SPC [19]. Applicability, with some adjustments, has been demonstrated on nonlinear systems [28], [29], [40].

Using the data-based system representation (12) as equality constraints, the DeePC optimization problem to be solved at every time step can be stated as

$$\min_{g, u, y, \sigma} \sum_{k=1}^N \|y_k - r_k\|_Q^2 + \sum_{k=0}^{N-1} \|\Delta u_k\|_R^2 + \lambda_g h(g) + \lambda_y \|\sigma\|_2^2 \quad (16a)$$

$$\text{s.t. } \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma \\ 0 \\ 0 \end{bmatrix} \quad (16b)$$

$$\Delta u_k \in \Delta \mathcal{U} \quad \forall k \in \{0, \dots, N-1\} \quad (16c)$$

$$u_k \in \mathcal{U} \quad \forall k \in \{0, \dots, N-1\} \quad (16d)$$

$$y_k \in \mathcal{Y} \quad \forall k \in \{1, \dots, N\} \quad (16e)$$

where we penalize changes in the input and the deviation of the outputs from the reference to achieve smooth input trajectories.

The regularization $h(g)$ in (16a) leads to robustification of the predictive control problem with respect to inexact data arising from noise or nonlinear processes. Depending on the assumptions of the given data and the data generating process, different regularizers have been proposed and theoretically investigated [24]. In norm-based regularization, the p -norm of the optimization variable g is additionally penalized in the cost function, e.g., $h(g) = \|g\|_p$. This approach has initially been proposed in [21]. Using the one-norm has been derived as a convex relaxation of the low-rank approximation of the data matrix [24]. The two-norm relates to the preconditioning of the predictor in an SPC sense [48]. The Tikhonov regularizer $h(g) = \|g\|_2^2$, together with terminal equilibrium constraints, has been used for proving closed-loop stability and providing robustness guarantees [31], [40]. Alternatively, the orthogonal projector on the kernel of the first three block-constraint equations $(I - \Pi)g$, where

$$\Pi = \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \quad (17)$$

has been proposed and robustification akin to least-square identification is stated, i.e., $\|(I - \Pi)g\| = 0$ if and only if the least-squares criterion is minimized [24], [25]. The main difference between norm-based and projection-based regularization is consistency. Norm-based regularizers are not consistent since they penalize the heterogeneous solution of the equality constraints in (16). They introduce a bias to the solution of the predictive control problem that depends on the magnitude of the regularization parameter λ_g . Projection-based regularizers, on the other hand, only penalize the homogeneous solution and are, therefore, consistent.

Furthermore, since the constraints (16b) used to obtain the initial condition of the future trajectory can be affected by stochastic disturbance, it may not be feasible in such settings [22]. Therefore, the strict equality is relaxed using a slack variable $\sigma \in \mathbb{R}^{pT_{\text{ini}}}$ which is penalized in the cost function. Choosing $\lambda_y \|\sigma\|_2$ results in a least-square-type initial

condition estimate reminiscent of moving horizon estimation [23], [49].

We include box constraints for the change in inputs Δu , the absolute values of the inputs u , and the outputs y . To guarantee feasibility of (16), we additionally soften the output constraints using slack variables which are heavily penalized in the ∞ -norm weighted by 10^4 to recover the original solution if feasible. It is important to note that with an appropriate choice of the weighting matrices Q and R , the optimization problem (16) is a convex QP. This facilitates the embedded optimization procedure and established numerical QP solvers can be used.

The equality constraints (16b) can be used to eliminate u , y , and σ from problem (16). The resulting condensed DeePC formulation with g as the only remaining free variable reads as

$$\min_g \sum_{k=0}^N (\|Y_f g - y_r\|_Q^2 + \|U_f K_u u\|_R^2) + \lambda_g h(g) + \lambda_y \|Y_p g - y_{\text{ini}}\|_2^2 \quad (18a)$$

$$\text{s.t. } U_p g = u_{\text{ini}} \quad (18b)$$

$$u_k \in \mathcal{U} \quad \forall k \in \{0, \dots, N-1\} \quad (18c)$$

$$y_k \in \mathcal{Y} \quad \forall k \in \{1, \dots, N\} \quad (18d)$$

where K_u has 1s on the main diagonal and -1 on the first lower diagonal to penalize the input changes $\Delta u = u_k - u_{k-1}$. Note that compared with [25, Eq.(44)], we do not relax the first block of equality constraints, reflecting the assumption that there is no uncertainty in the (past) input signals. Therefore, this equality remains in the condensed problem formulation (18).

F. Numerical Solution on Embedded Hardware

The optimization problem (16) is solved using derivative-based optimization routines. To exploit the sparsity structure of the QP, we use the open-source C code solver operator splitting QP (OSQP) based on the direct single-threaded linear system solver QDLDL [50]. The symbolic factorization is calculated offline. The solver is code-generated and compiled for the target. For comparison, we use the established parametric active-set solver qpOASES, originally developed for small- to medium-scale QPs arising from MPC, for the condensed formulation (18) [51].

All the reported turnaround times are the result of HiL experiments on the dSPACE MicroAutoBox II (MABX II). The MABX II is equipped with the DS1401 board containing a single-core PowerPC 750 GL processor running at 900 MHz and 16-MB flash memory. Computational costs are evaluated using the turnaround time t_c , which is defined as the total computation time needed on the embedded hardware to deterministically complete the calculation of the next input signal.

IV. APPLICATION TO A FUEL CELL SYSTEM

We apply algorithms (16) and (18) with a sampling time of $T_s = 0.2$ s to the fuel cell system simulation. We empirically set $Q = \text{diag}([0, 1, 10])$, $R = 0.1 I$, and $N = 5$, where the surge distance is not tracked but only constrained. According to Q , a higher priority is assigned to the net power output

tracking compared with stoichiometry tracking. In contrast to MPC, DeePC requires a past horizon T_{ini} to determine the current system state and the system representation g . In all our experiments, the length of the past horizon is set to $T_{\text{ini}} = 5$. A longer past horizon did not prove beneficial, while a shorter past horizon increases the closed-loop cost. Tuning of λ_g addresses the tradeoff of matching the linear column combination g to the past T_{ini} data points or the potentially unreachable future reference. We observe more conservative controller tuning, i.e., constraint satisfaction, with $\lambda_g \gg 1$ and more aggressive controller tuning, i.e., faster reference tracking but some constraint violations, with $\lambda_g \ll 1$. Intuitively, with increasing λ_g , the flexibility of the system representation is more restricted. The slack variable σ is penalized using $\lambda_y = 10^4$ to achieve accurate system identification on past data without adversely affecting the conditioning of the optimization problem too much.

A. DeePC With Fixed Data Matrix

For our first experiments, we assume an initial data trajectory of length $T = 970$ from which $n_c \in [100, 400]$ columns are selected offline according to the CSS algorithms. These data stem from 190-s closed-loop control using a two-degree-of-freedom (2DoF) controller consisting of static feedforward maps and single-input single-output control loops with PID controllers. The resulting data matrix is then normalized and fixed during the closed-loop experiments.

1) *Column Subset Selection*: The CSS algorithms are applied to the Hankel matrix formed from the initial data trajectory. A visualization of the input and output data trajectory and the starting index of selected columns are depicted in Fig. 3. With norm sampling, only a few parts of the initial trajectory are sampled densely, while large parts are not considered. According to this sampling algorithm applied to the z -score normalized data matrix, the extreme input signals lead to high norms and are therefore selected. Columns similar to each other are selected multiple times, such that only a few operating points are represented in the data matrix. Leverage score, iterative norm, and sRRQR selection qualitatively select trajectories that are more spread out across the operating range of the initial data trajectory. Furthermore, the transient parts are preferred by those CSS algorithms, which can be expected to contain more information than similar stationary parts.

We evaluate the closed-loop performance of the DeePC algorithm for the different CSS algorithms and a varying number of columns n_c . For comparison, 20 closed-loop simulations with randomly collected columns are conducted for each n_c . Fig. 4 summarizes the closed-loop cost according to (4) followed by normalization for all the configurations. With an increasing number of columns n_c , the system representation tends to improve such that the prediction becomes increasingly accurate and the overall closed-loop costs are reduced for all the investigated CSS algorithms and the randomly selected columns. For all $n_c \in [100, 400]$, the sRRQR algorithm leads to the lowest closed-loop cost, followed by leverage score sampling and iterative norm sampling with approximately equal results. In particular, those CSS algorithms yield consistently lower closed-loop costs than the first quartile of random

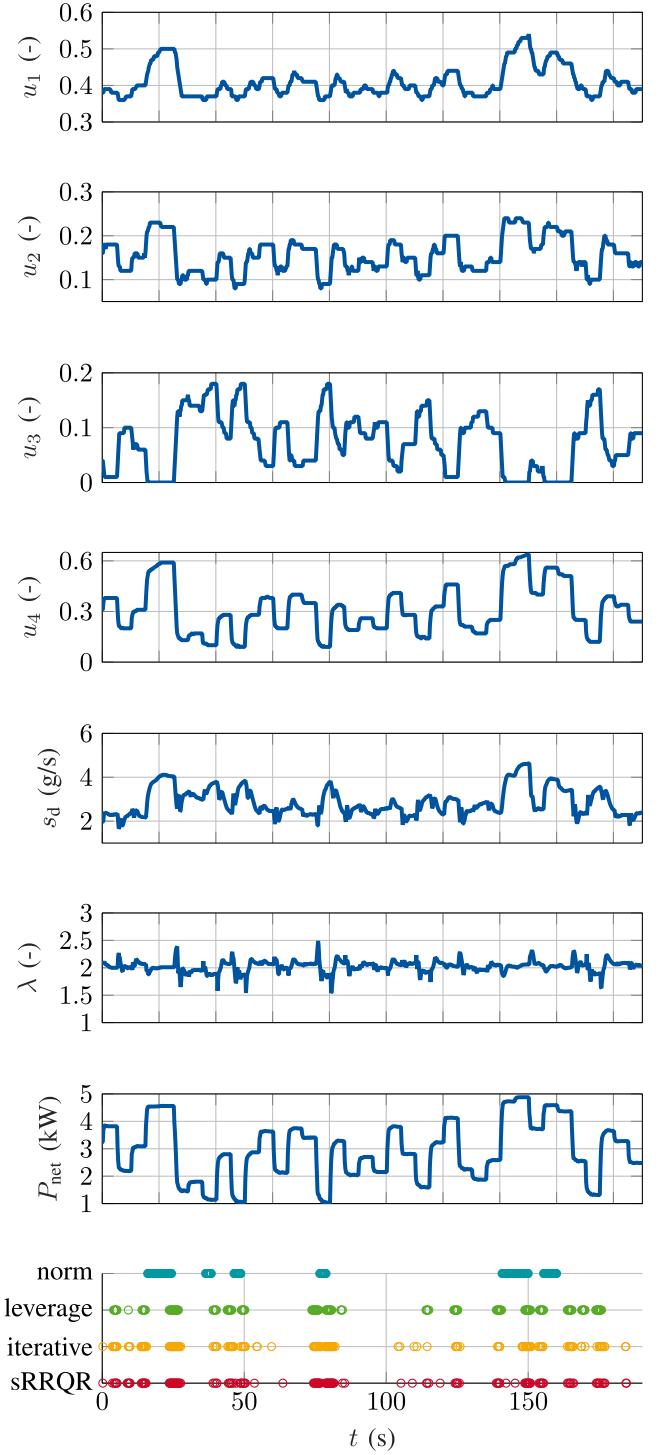


Fig. 3. Initial input and output trajectories used for evaluating the data selection algorithms. The start of each selected trajectory of length $L = T_{\text{ini}} + N = 10$ is marked in the bottom plot. The selection of $n_c = 150$ columns is marked in the bottom plot.

column selection. In addition, the spread in closed-loop cost for random selection is high, such that especially for small n_c , the performance varies strongly. Increasing the number of columns to $n_c \geq 350$, the differences between random sampling and the CSS algorithms in closed loop become small. With norm sampling, only a few parts of the initial trajectory are sampled densely, while large parts are not considered. This sampling algorithm selects columns with extreme input

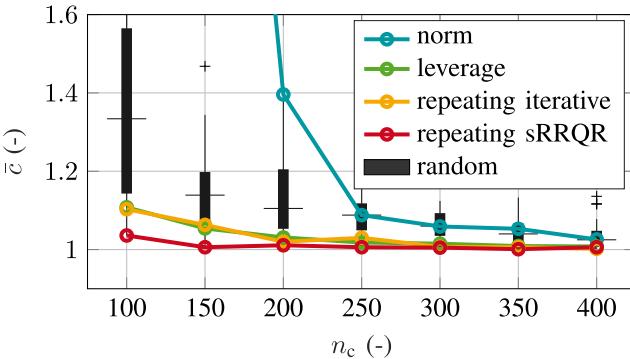


Fig. 4. Normalized closed-loop tracking cost for varying n_c and different CSS algorithms. For comparison, 20 simulations with random columns selected are summarized in the boxplot.

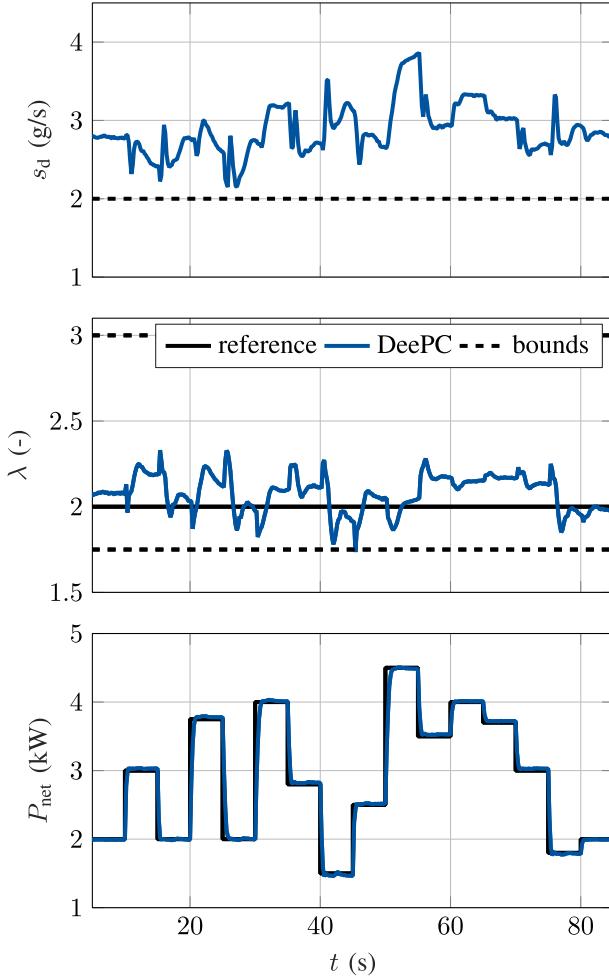


Fig. 5. Closed-loop control result with DeePC using an offline selected data matrix using sRRQR sampling with $n_c = 150$.

signals resulting in a large norm. Columns that are similar to each other are selected multiple times, such that only a few operating points are represented in the data matrix. This leads to overfitting to a few operating points, impaired tracking, and increased closed-loop cost. For online adaption of columns, both norm sampling and random adaption can be excluded because of unsatisfactory or high-variance closed-loop performance.

2) *Control Results:* For sRRQR sampling and $n_c = 150$ columns, the closed-loop simulation results are shown in

TABLE I
CLOSED-LOOP COST COMPARISON FOR ALL CONTROLLERS

Controller	T_s	$\ y - r\ _Q^2$	$\ \Delta u\ _R^2$	c
DeePC	200 ms	4.58	0.14	4.72
MPC	50 ms	3.12	0.08	3.20
2DoF	50 ms	6.50	0.03	6.52

Fig. 5 in the time domain for a power step reference cycle with a desired constant stoichiometry $\lambda = 2$. The net power output and the stoichiometry are tracked closely even during the transients of steps in the reference trajectory. The operational constraints on the stoichiometry and the surge distance are respected at all times. Interestingly, we achieved close-to-identical results with different regularization schemes after manually retuning the penalty $\lambda_g \in [1, 20]$ to avoid constraint violations. From the perspective of closed-loop control, those methods are therefore equivalent regarding our application. Furthermore, a system representation for predictive control with $n_c = 150$ can be found purely based on data without explicitly identifying a state-space model.

3) *Comparison to MPC and Conventional Control:* We compare DeePC to nonlinear MPC and the conventional controller (2DoF) used to create the initial data trajectory. We impose a real-time constraint for all the algorithms on the given embedded hardware, i.e., all the algorithms are validated in HiL experiments. We implement nonlinear MPC based on an identified nonlinear state-space model and a prediction horizon of $N = 10$. The resulting Nonlinear Program is solved using sequential quadratic programming (SQP) with a constant step length, a single SQP iteration per time step, the Gauss–Newton Hessian approximation, condensing, and qpOASES as the sub-QP solver. In addition, we implement state and disturbance estimation for offset-free control in stationary operation. Relying on the mature algorithmic development for MPC, we achieve a worst case turnaround time of $\max(t_c) = 22$ ms such that we can choose the algorithm's sampling time to $T_s = 50$ ms.

The 2DoF controller also operates with $T_s = 50$ ms at negligible computational cost. Note that this sampling time is considerably shorter than for DeePC. Despite the effort to find a concise system representation, the problem dimensions of DeePC restrict the sampling time to $T_s = 200$ ms. For comparability of the closed-loop cost, we evaluate (4) every 50 ms for all the controllers. The sampling times and the resulting closed-loop cost for all the controllers are summarized in Table I. The tracking costs dominate, as they are between one and two orders of magnitude larger than the actuation penalties for all the controllers and reference cycles. We did not encounter excessive changes or oscillations in the actuated inputs such that a comparatively small weight is deemed sufficient. MPC results in the lowest closed-loop cost on our reference cycle (3.20), followed by DeePC (4.72) and 2DoF control (6.52). While constraints can be satisfied with all the controllers, MPC achieves the fastest net power reference tracking. A precise model of the physical system allows to accurately predict its behavior and fully leverage the potential

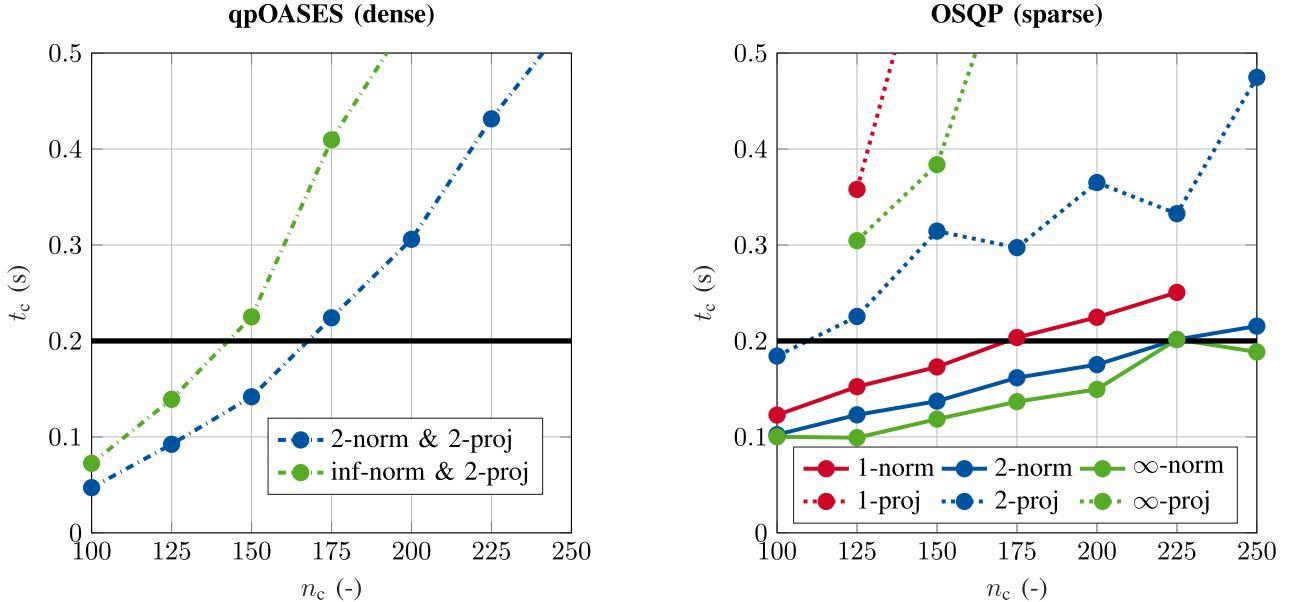


Fig. 6. Turnaround time on embedded hardware board for norm-based (norm) and projection-based (proj) regularization using the one-, two-, and ∞ -norm where possible. Problem (18) with qpOASES (left) and problem (16) with OSQP (right). Missing data points lead to failed simulations. Only results with turnaround time smaller or equal to 0.5 s are depicted. The turnaround time must be strictly below 0.2 s (black line) to achieve real-time feasibility.

of model-based control. DeePC cannot be deployed with the same fast sampling time as MPC and we need to increase its sampling time to 200 ms for real-time feasibility with our configuration. With a one-time step delay for computation, this inevitably affects disturbance rejection and delays reference tracking. Furthermore, noisy data must be addressed by regularization, which further slows down the closed-loop response. Compared with the conventional 2DoF control, closed-loop costs can be reduced significantly.

It should be noted that DeePC allows a reduction in implementation effort compared with MPC by substituting a sufficiently rich data trajectory for the need to identify a system model. Furthermore, the resulting optimization problem is a QP that can be reliably solved using open-source solver implementations. However, with DeePC still in its infancy, accelerating online optimization and thus reducing the sampling time is imperative to make the algorithm competitive in the future.

4) Computational Results on Embedded Hardware:

The number of optimization variables, equality constraints, and inequality constraints is identical for norm-based and projection-based regularization. The sparsity pattern and the number of nonzero entries in the QP matrices, however, change. For the example of squared two-norm regularization for $n_c = 100$, the number of nonzero entries (nnz) in the Hessian is increased from nnz = 177 to nnz = 10077 when using the projection-based regularization compared with the norm-based regularization. Norm-based regularization adds a (sparse) diagonal $I \in \mathbb{R}^{n_c \times n_c}$ block, while $(I - \Pi) \in \mathbb{R}^{n_c \times n_c}$ is densely populated. This heavily impacts numerical solvers based on sparse linear algebra due to additional fill-in in the factorization. For the OSQP solver, not only the worst case runtime more than doubles, see Fig. 6, but also the size of the generated code containing the offline calculated sparse matrix

factorizations increases by more than 50% for projection-based regularization. The increased number of nonzero entries prohibits a real-time feasible solution for all the tested n_c except for the two-norm projection-based regularization with $n_c = 100$. For sparse, norm-based regularization, we observe approximately linear complexity in the optimization variables. For $n_c \leq 200$, two-norm and ∞ -norm regularization lead to real-time feasible controllers. Relating this result with Fig. 4, the concise system representation is necessary for real-time embedded optimization. The condensed DeePC problem is solved with an active-set method (qpOASES). Regularization with the two-norm and the ∞ -norm leads to real-time feasible controllers for $n_c \leq 150$ and $n_c \leq 125$, respectively. Due to an increase in optimization variable when reformulating the one-norm regularized optimization problem, worst case runtime increases for qpOASES to more than 0.5 s and is therefore not depicted in Fig. 6. Considering the higher solution accuracy of an active-set method in the order of 10^{-7} compared with the default settings of OSQP with $\varepsilon_{\text{rel}} = \varepsilon_{\text{abs}} = 10^{-3}$, we do not observe any visible difference in closed-loop control. This indicates that a high-accuracy solution is not necessarily needed, similar to the results in the MPC domain [52]. In summary, the two-norm regularization is computationally advantageous and will be used in the following in the sparse problem formulation solved with OSQP.

B. DeePC With Adaptive Data Matrix

In this section, the data matrix with $n_c = 200$ is divided into a part with fixed columns selected offline in the beginning and a part with interchangeable columns updated online according to Algorithm 1. For the division, an 80/20 split of fixed and free trajectories is chosen. As all the CSS algorithms select the most informative columns first, simply 80% of the columns sampled first are fixed and the remaining 20% are set as free

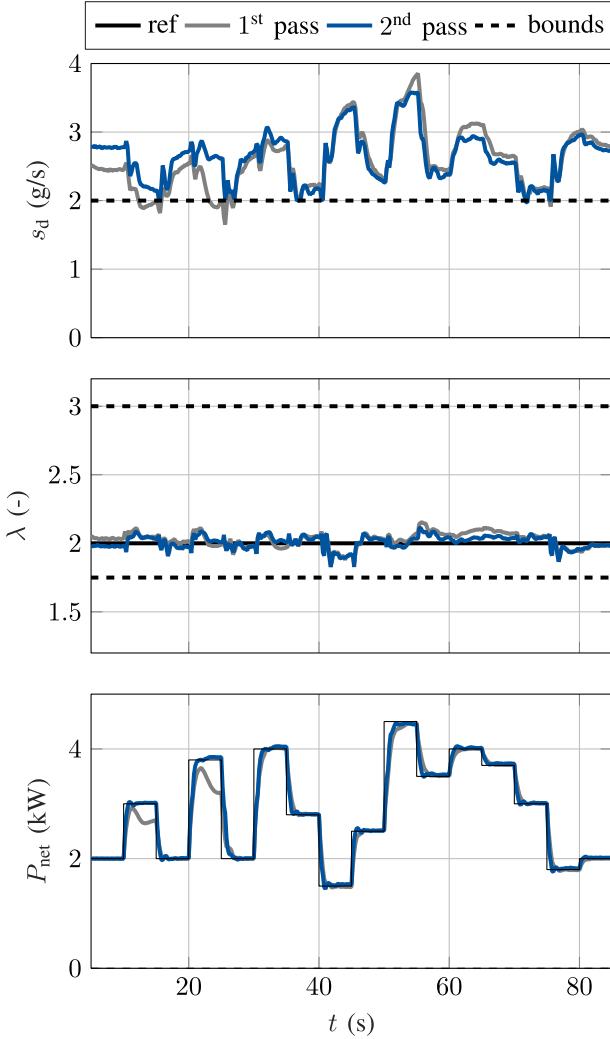


Fig. 7. Closed-loop simulation without data for net power $P_{\text{net}} \geq 2.5$ kW before and after applying the online adaption in the second pass through the reference cycle.

trajectories. Memory restrictions on the hardware prohibit the exchange of more trajectories. We use leverage score sampling for the initial selection and online adaption.

1) *Control Results*: We evaluate the online update procedure of the data matrix in a scenario where the required operating regimes are not (yet) represented in the data matrix. Therefore, before running the initial CSS algorithm on the Hankel matrix, all the columns containing data points with a net power $P_{\text{net}} \geq 2.5$ kW are removed such that no information for high power requests is contained in the data matrix. Then, the DeePC algorithm is set up analogously to offline setting and a test cycle is simulated. During that cycle, the online adaption of the data matrix is active, but only after the first cycle, the data matrix in DeePC is updated. Thus, for the second pass through the test cycle, the DeePC algorithm can use the trajectories selected from the first repetition to make more accurate predictions into the future and improve the tracking performance, especially for transient operation with high net power requests. The closed-loop results are depicted in Fig. 7, where we overlay both cycle passes for better visual inspection of the difference. The closed-loop tracking performance deteriorates for higher net power requests that

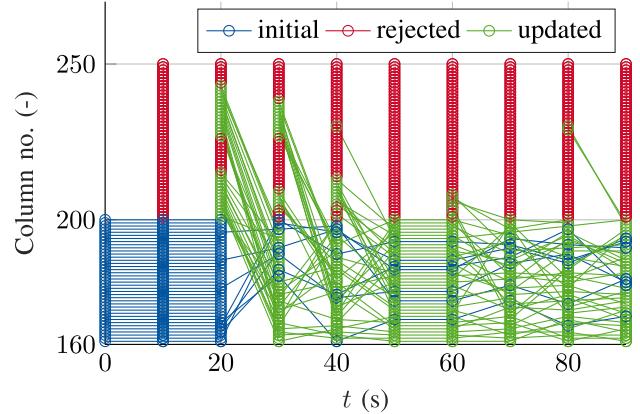


Fig. 8. Column selection over time with interchangeable columns from no. 160 to 200 and newly collected columns from no. 201 to 250. The initial columns of the data matrix are marked in blue, online collected but rejected columns in red, and online collected and selected columns in green.

are not contained in the data matrix during the first pass. This indicates that in contrast to the linear case, the operating regime must be covered well for a general nonlinear system. Even though tracking is imprecise in the first pass and constraints are slightly violated, trajectories containing information about high-load operating points are collected and saved in the updated data matrix. Once the update gets fed back to the DeePC algorithm after the first pass, tracking performance improves, measured by a reduction of 25% in closed-loop tracking cost. In addition, in the second run of the cycle, no constraint violations of the minimum surge distance are reported.

A visualization of the data adaption over time is depicted in Fig. 8. Every column that can be selected is marked as a dot at the corresponding column number. Every $T_a = 10$ s, the criterion triggering a data update is checked. If the criterion is fulfilled, the 40 interchangeable and 50 newly generated data columns are combined and leverage score sampling is used to select 40 columns from the whole set. The line plots indicate which columns are selected over time. Online collected but rejected columns are marked in red, columns from the initial data matrix are marked in blue, and columns collected and selected online are marked in green. From the initial data matrix, almost all the columns that can be interchanged are exchanged during the first pass through the cycle. Furthermore, by comparing Figs. 7 and 8, it can be seen that mostly data covering high-load operating regions are selected. The collected columns additionally cover the most dynamic operating points shortly after changes in the reference occurring every 5 s, which corresponds to columns 201 and 226.

2) *Computational Results on Embedded Hardware*: Using the adaption algorithm leads to additional memory requirements and an increase in computational load. While the DeePC algorithm is still running at $T_s = 0.2$ s, Algorithm 1 is executed with a sampling time of $T_a = 10$ s. Since the adaption algorithm is assigned a lower priority on the single-core board, it is carried out in the idle phase of the DeePC algorithm. For this scenario, we report the turnaround times for both the tasks in Table II for the two-norm regularization on g and using the OSQP solver for (16). Both the tasks stay well

TABLE II

TURNAROUND TIMES FOR POWER REFERENCE CYCLE				
Algorithm	T_s	$\max(t_c)$	$\min(t_c)$	$\text{mean}(t_c)$
DeePC	0.2 s	0.18 s	0.06 s	0.07 s
Algorithm 1	10 s	2.4 s	0.1 s	0.86 s

below their respective sampling time, and the overall CPU time is sufficient for both the tasks. We conclude that the presented update procedure in combination with DeePC is real-time feasible on the given hardware board.

C. Discussion

The behavioral system representation in predictive control leads to satisfactory closed-loop control performance for the nonlinear fuel cell system if the operating range is covered reasonably well in the data. Data adaption can safeguard against unexplored areas and update the data matrix accordingly. In Fig. 5, one can observe that data-based control not necessarily leads to offset-free control. Again, the adaption of the data matrix can alleviate this issue. Besides the data matrix, closed-loop cost and constraint satisfaction rely on the hyper-parameter λ_g . Choosing λ_g too small leads to more and more severe constraint violations by allowing unrealistic model flexibility, while choosing it too high impedes closed-loop tracking performance. An acceptable tradeoff must be found.

From an embedded optimization point of view, solving the DeePC optimization problem is challenging due to a large number of optimization variables required by the system representation. Numerical solution methods based on sparse linear algebra benefit from the sparsity of the norm-based regularization scheme in terms of runtime and required memory. Established algorithms in MPC based on dense linear algebra perform better on the condensed optimization problem. Real-time feasibility and real-world applicability for all the methods, however, depends on a concise system representation and small problem dimensions m and p . Dedicated optimization algorithms, e.g., exploiting block-Hankel structure through matrix–vector multiplication using a Fast Fourier Transform [39], exist but rely on specific structure lost through CSS.

V. CONCLUSION

This work illustrates the practical aspects of using DeePC in real-world applications. A concise system representation shown to be real-time feasible with satisfactory closed-loop performance is found with CSS algorithms. An update algorithm is developed and validated in a simulation study leading to a consistent reduction in closed-loop cost. Experiments on embedded hardware indicate real-time feasibility of all the presented algorithms. Future work will focus on the validation of the proposed scheme using a fuel cell system test bed.

ACKNOWLEDGMENT

The authors would like to thank Sönke Gössling, Michael Schoemaker, and Niklas Nickig from the hydrogen fuel cell center ZBT GmbH for providing the fuel cell simulation and

AVL List GmbH for providing the simulation software. The research project was carried out in the framework of the industrial collective research program (IGF no. 61 EWN).

REFERENCES

- [1] L. Guzzella and C. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems*, Berlin, Germany: Springer, 2013.
- [2] J. T. Pukrushpan, A. G. Stefanopoulou, and H. Peng, *Control of Fuel Cell Power Systems: Principles, Modeling, Analysis and Feedback Design*. London, U.K.: Springer-Verlag, 2004.
- [3] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. San Francisco, CA, USA: Nob Hill Publishing LLC, 2019.
- [4] A. Arce, A. J. del Real, C. Bordons, and D. R. Ramirez, “Real-time implementation of a constrained MPC for efficient airflow control in a PEM fuel cell,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 6, pp. 1892–1905, Jun. 2010.
- [5] M. A. Danzer, S. J. Wittmann, and E. P. Hofer, “Prevention of fuel cell starvation by model predictive control of pressure, excess ratio, and current,” *J. Power Sources*, vol. 190, no. 1, pp. 86–91, May 2009.
- [6] M. Vrlić, D. Ritzberger, and S. Jakubek, “Safe and efficient polymer electrolyte membrane fuel cell control using successive linearization based model predictive control validated on real vehicle data,” *Energies*, vol. 13, no. 20, p. 5353, Oct. 2020.
- [7] R. Nebeluk and M. Ławryńczuk, “Fast model predictive control of PEM fuel cell system using the L_1 norm,” *Energies*, vol. 15, no. 14, p. 5157, Jul. 2022.
- [8] V. Neisen, J. Mannhardt, and D. Abel, “Dynamic tracking of power demand for integrated fuel cell systems using nonlinear model predictive control,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 13216–13223, 2020.
- [9] J. K. Gruber, C. Bordons, and A. Oliva, “Nonlinear MPC for the airflow in a PEM fuel cell using a Volterra series model,” *Control Eng. Pract.*, vol. 20, no. 2, pp. 205–217, Feb. 2012.
- [10] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
- [11] L. Ljung, “Identification for control: Simple process models,” in *Proc. 41st IEEE Conf. Decis. Control*, Oct. 2002, pp. 4652–4657.
- [12] A. Saengrung, A. Abtahi, and A. Zilouchian, “Neural network model for a commercial PEM fuel cell system,” *J. Power Sources*, vol. 172, no. 2, pp. 749–759, Oct. 2007.
- [13] J. T. Pukrushpan, H. Peng, and A. G. Stefanopoulou, “Control-oriented modeling and analysis for automotive fuel cell systems,” *J. Dyn. Syst., Meas., Control*, vol. 126, no. 1, pp. 14–25, Mar. 2004.
- [14] A. Vahidi, A. Stefanopoulou, and H. Peng, “Current management in a hybrid fuel cell power system: A model-predictive control approach,” *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 6, pp. 1047–1057, Nov. 2006.
- [15] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems. Theory, Implementation, Applications*, vol. 14. Cham, Switzerland: Springer, 1996.
- [16] B. De Moor and M. Gevers, “SPC: Subspace predictive control,” in *Proc. 14th IFAC World Congr.*, 1999, vol. 32, no. 2, pp. 1–15.
- [17] R. Hallouzi and M. Verhaegen, “Persistency of excitation in subspace predictive control,” in *Proc. 17th IFAC World Congr.*, 2008, vol. 41, no. 2, pp. 11439–11444.
- [18] B. Huang and R. Kadali, *Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-Driven Subspace Approach*. Cham, Switzerland: Springer, 2008.
- [19] F. Fiedler and S. Lucia, “On the relationship between data-enabled predictive control and subspace predictive control,” in *Proc. Eur. Control Conf. (ECC)*, Jun. 2021, pp. 222–229.
- [20] H. Yang and S. Li, “A data-driven predictive controller design based on reduced Hankel matrix,” in *Proc. 10th Asian Control Conf. (ASCC)*, May 2015, pp. 1–7.
- [21] J. Coulson, J. Lygeros, and F. Dörfler, “Data-enabled predictive control: In the shallows of the DeePC,” in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 307–312.
- [22] J. Coulson, J. Lygeros, and F. Dörfler, “Regularized and distributionally robust data-enabled predictive control,” in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 2696–2701.
- [23] J. Coulson, J. Lygeros, and F. Dörfler, “Distributionally robust chance constrained data-enabled predictive control,” *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3289–3304, Jul. 2022.

- [24] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct and indirect data-driven control formulations via regularizations and relaxations," *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 883–897, Feb. 2023.
- [25] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annu. Rev. Control*, vol. 52, pp. 42–64, Jan. 2021.
- [26] J. C. Willems and J. W. Polderman, *Introduction to Mathematical Systems Theory: A Behavioral Approach*, vol. 26. Cham, Switzerland: Springer, 1997.
- [27] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor, "A note on persistency of excitation," *Syst. Control Lett.*, vol. 54, no. 4, pp. 325–329, Apr. 2005.
- [28] E. Elokdja, J. Coulson, P. N. Beuchat, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for quadcopters," *Int. J. Robust Nonlinear Control*, vol. 31, no. 18, pp. 8916–8936, Dec. 2021.
- [29] P. G. Carlet, A. Favato, S. Bolognani, and F. Dörfler, "Data-driven continuous-set predictive current control for synchronous motor drives," *IEEE Trans. Power Electron.*, vol. 37, no. 6, pp. 6637–6646, Jun. 2022.
- [30] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for grid-connected power converters," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 8130–8135.
- [31] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Quadratic regularization of data-enabled predictive control: Theory and application to power converter experiments," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 192–197, 2021.
- [32] Y. Lian, J. Shi, M. Koch, and C. N. Jones, "Adaptive robust data-driven building control via bilevel reformulation: An experimental result," *IEEE Trans. Control Syst. Technol.*, early access, Mar. 30, 2023, doi: 10.1109/TCST.2023.3259641.
- [33] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Robust data-enabled predictive control: Tractable formulations and performance guarantees," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 3163–3170, May 2023.
- [34] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021.
- [35] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Robust constraint satisfaction in data-driven MPC," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 1260–1267.
- [36] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "On the design of terminal ingredients for data-driven MPC," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 257–263, 2021.
- [37] J. Bongard, J. Berberich, J. Köhler, and F. Allgöwer, "Robust stability analysis of a simple data-driven model predictive control approach," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2625–2637, May 2023.
- [38] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Linear tracking MPC for nonlinear systems—Part II: The data-driven case," *IEEE Trans. Autom. Control*, vol. 67, no. 9, pp. 4406–4421, Sep. 2022.
- [39] S. Baros, C.-Y. Chang, G. E. Colón-Reyes, and A. Bernstein, "Online data-enabled predictive control," *Automatica*, vol. 138, Apr. 2022, Art. no. 109926.
- [40] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control: Closed-loop guarantees and experimental results," *Automatisierungstechnik*, vol. 69, no. 7, pp. 608–618, Jul. 2021.
- [41] L. Schmitt, J. Beerwerth, and D. Abel, "Data selection and data-enabled predictive control for a fuel cell system," in *Proc. IFAC World Congr.*, 2023, pp. 4842–4847.
- [42] S. Gößling, N. Nickig, and M. Bahr, "2-D + 1-D PEM fuel cell model for fuel cell system simulations," *Int. J. Hydrogen Energy*, vol. 46, no. 70, pp. 34874–34882, Oct. 2021.
- [43] S. Gößling, "2-D + 1-D ortsaufgelöste modellierung von pem-brennstoffzellen," Ph.D. thesis, Dept. Mech. Process Eng., Universität Duisburg-Essen, Duisburg, Germany, 2019.
- [44] J. Berberich and F. Allgöwer, "A trajectory-based framework for data-driven system analysis and control," in *Proc. Eur. Control Conf. (ECC)*, 2020, pp. 1365–1370.
- [45] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *Int. J. Control.*, vol. 81, no. 12, pp. 1946–1959, Dec. 2008.
- [46] Y. Wang and A. Singh, "An empirical comparison of sampling techniques for matrix column subset selection," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 1069–1074.
- [47] C. Boutsidis, P. Drineas, and M. Magdon-Ismail, "Near-optimal coresets for least-squares regression," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6880–6892, Oct. 2013.
- [48] A. Xue and N. Matni, "Data-driven system level synthesis," *Proc. Mach. Learn. Res.*, vol. 144, pp. 1–12, Jan. 2021.
- [49] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
- [50] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Math. Program. Comput.*, vol. 12, pp. 637–673, Feb. 2020.
- [51] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "QpOASES: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, Dec. 2014.
- [52] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.



Lukas Schmitt received the B.S. and M.S. degrees in mechanical engineering from RWTH Aachen University, Aachen, Germany, in 2016 and 2019, respectively, where he is currently pursuing the Ph.D. degree in control engineering with the Institute of Automatic Control.

His research interests include model-based and data-based predictive control, embedded optimization, and their application to problems in the energy, aerospace, and automotive domains.



Julius Beerwerth received the B.S. and M.S. degrees in computational engineering science from RWTH Aachen University, Aachen, Germany, in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree in computer science with the Chair of Embedded Software.

During the master's thesis, he worked on the online update algorithm presented in this work. His research interests include embedded software architecture, predictive control, and optimization of autonomous vehicles.



Matthias Bahr received the B.S. and M.S. degrees in mechanical engineering from the University of Duisburg-Essen, Duisburg, Germany, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree in mechanical engineering with the Chair of Mechatronics, University of Duisburg-Essen, in collaboration with The hydrogen and fuel cell center ZBT GmbH, Duisburg.

His research interests include combining machine learning algorithms with modeling and control of fuel cell systems.



Dirk Abel received the Dipl.-Ing. degree in mechanical engineering and the Ph.D. degree from RWTH Aachen University, Aachen, Germany, in 1983 and 1987, respectively.

In 2001, he became a Full Professor and the Head of the Institute of Automatic Control, RWTH Aachen University. From 2009 to 2017, he was a member of the Supervisory Board of KUKA AG, Augsburg, Germany. His main research interests include model predictive control, nonlinear and robust control, and rapid control prototyping in different application areas.