Real-Time Feasibility of Data-Driven Predictive Control for Synchronous Motor Drives

Journal Article

Author(s):
Carlet, Paolo Gherardo; Favato, Andrea; Torchio, Riccardo; Toso, Francesco; Bolognani, Saverio; Dörfler, Florian

Publication date:
2023-02

Permanent link:
https://doi.org/10.3929/ethz-b-000579513

Originally published in:
IEEE Transactions on Power Electronics 38(2), https://doi.org/10.1109/tpel.2022.3214760

Published on the IEEE Transactions on Power Electronics (2022).
https://doi.org/10.1109/TPEL.2022.3214760

Real-Time Feasibility of Data-Driven Predictive Control for Synchronous Motor Drives

Paolo Gherardo Carlet, Andrea Favato, Riccardo Torchio, Francesco Toso, Saverio Bolognani, Florian Dörfler

***Abstract*—The data-driven control paradigm allows overcoming conventional troubles in the controller design related to model identifications procedures. Raw data are directly exploited in the control input selection by forcing the future plant dynamics to be coherent with previously collected samples. This paper focuses, in particular, on the data-enabled predictive control algorithm. A relevant disadvantage of this algorithm is the fact that the complexity of the online control program grows with the dimension of the data-set. This issue becomes particularly relevant when considering embedded applications such as the control of synchronous motor drives, characterized by challenging real-time constraints. This work proposes a systematic approach for dramatically reducing the complexity of such algorithms. Such methodology enables real-time feasibility of the constrained version of this control structure, which was previously precluded. Simulations and experimental results are provided to validate the method, considering the current control of an interior permanent magnet motor as test-case.**

***Index Terms*—Data-enabled predictive control (DeePC), model predictive control (MPC), permanent magnet synchronous motor (PMSM), proper orthogonal decomposition (POD)**

I. Introduction

Digitalization of industrial processes is generating huge amount of data day by day. This revolution provides unprecedented opportunities for engineers and practitioners [1], since devices and plants can be monitored and analyzed in every moment of their life-cycle.

In the control engineering community, a fresh data-based perspective on control is emerging [2], parallel to the most conventional model-based approaches. In particular, given the richness of information about dynamics, data-driven control design procedures appear convenient.

Among data-driven controllers, the data-enabled predictive control (DeePC) is considered in this work [3]. Being a predictive controller, the idea behind DeePC is to optimize future plant inputs to minimize a defined cost function, e.g., a tracking control problem, in a similar manner as a model predictive control (MPC). Most importantly, input and output constraints can be included in the optimization. However, the plant output prediction is performed by means of data previously collected from the plant, instead of conventional state-space models.

DeePC has already found applications in different areas, including aerial robotics, automotive, power electronics and power systems [4], characterized by different timescales. In this work, the power electronics area is considered, one of the most challenging because of a very tight timescale. Considering such area, predictive controllers are categorized into finite-control-set methods and continuous-control-set methods [5]. DeePC refers to latter category. Thus, DeePC computes the equivalent voltage vectors feeding the motor windings. Then, a pulse width modulation algorithm finds the duty cycles of the power converter switches. Differences and comparisons between finite-control-set and continuous-control-set predictive controller are discussed in [6].

The transition from model-based to data-driven design of predictive regulator for the current control of synchronous motor drives is described in [7]. This case study revealed a practical issue of such control strategy. In fact, the real-time constraints precluded the implementation of a constrained version of the algorithm. In particular, the optimal voltage to drive the machine was computed while neglecting the feasible voltage region of the power converter because of computation time limits.

The computation time issue was due to two main reasons:

- the lack of a customized solver for the control problem;
- data samples appear explicitly in the online program, and therefore its complexity grows with the amount of data used to replace the state-space model in the output prediction.

This paper aims to address both issues exploiting recent advances in customized solvers for embedded real-time optimization and well-know reduction techniques to handle the pre-collected data. As interesting feature of the proposed controller, no models are introduced to describe the dynamic, remaining faithful to the data-driven paradigm.

Many solvers recently have been proposed for embedded predictive control of power electronics. *Geyer et al.* have been developing efficient solvers for medium-high power electronics for one decade [8]. However, these methods find application only for finite-set predictive controllers [9] and cannot be generalized to MPC coupled with pulse-width-modulation [10], [11]. *Bemporad et al.* proposed an active set method for the current control of surface-mounted motors, certificating also the real-time feasibility [12]. Alternatively, a customized solver was proposed in [13], specifically designed for synchronous motors driven by three-phase two-levels converters. The latter method is considered here, since it shows promising real-time performances with respect to general purpose solver, such as qpOASES [14]. This method finds first the optimal unconstrained solution. If the solution is feasible, the algorithm stops. If not, the method start iterating among the violated constraints by considering them individually and exploiting the peculiarities of the specific control problem. This solver turns out to be particularly suitable for the application of standard reduction methods that allow for dramatically reducing the problem dimension and therefore the overall computational cost for solving the control problem.

The adopted reduction method is based on a classical proper orthogonal decomposition (POD) scheme [15], where the solution space is spanned in order to construct a proper basis that is used to project the problem according to a Galerkin scheme [16]. The problem reduction is implemented in the offline stage. Particular attention is also given to the stopping criterion adopted in the reduction algorithm in order to keep the reduced control problem small and well conditioned for both the unconstrained and constrained cases. The coupling between the considered solver and the problem reduction technique may result similar also to the explicit MPC approach [17].

In conclusion, this work proposes an effective methodology to reduce the computation time of the DeePC controller, applied to the current control of electric motor drives. The contributions of this work are manifold, and they can be summarized as:

- the computation burden of the DeePC is reduced by applying a POD technique directly to the control problem, without identifying any model;
- the reduction methodology is coupled to a customized MPC solver for electric motor drives application;
- the constrained DeePC is implemented for the first time on a dSPACE hardware at a rate of 10 kHz, demonstrating its real-time feasibility;
- the accuracy between the reduced DeePC problem solution and the original problem is studied;
- the differences between the constrained DeePC solution and the unconstrained program are analyzed.

We believe that our contributions ultimately enable to transfer a cutting-edge (albeit academic) data-driven control algorithm to practice. Moreover, the proposed methodology represents a key milestone towards adaptive DeePC, i.e. a controller capable to recursively self-adapt to the plant operating condition. In fact, a fast and effective procedure to handle raw data and build up a control law becomes essential for a recursive implementations with a continuous stream of data.

II. Theoretical Background

A. PMSM model

Before introducing the DeePC, a brief description is provided about the equations that model the permanent magnet synchronous motor (PMSM) currents dynamic. The most widespread model used to represent PMSM currents is based on the voltage balance equations, represented in the direct-quadrature (dq) reference frame, synchronous with the rotor flux. Since a digital controller is often implemented to drive the motor, the continuous time equations are discretized using the Euler method. The resulting model is reported below

$$\boldsymbol{i}_{\mathrm{dq}}(k+1) = \mathbf{A}\boldsymbol{i}_{\mathrm{dq}}(k) + \mathbf{B}\boldsymbol{u}_{\mathrm{dq}}(k) + \mathbf{B}\boldsymbol{h}(k)$$
$$\mathbf{A} = \begin{bmatrix} 1 - R_{\mathrm{s}}\dfrac{T_{\mathrm{s}}}{L_{\mathrm{d}}} & \omega_{\mathrm{e}}\dfrac{L_{\mathrm{q}}}{L_{\mathrm{d}}}T_{\mathrm{s}} \\ -\omega_{\mathrm{e}}\dfrac{L_{\mathrm{d}}}{L_{\mathrm{q}}}T_{\mathrm{s}} & 1 - R_{\mathrm{s}}\dfrac{T_{\mathrm{s}}}{L_{\mathrm{q}}} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \dfrac{T_{\mathrm{s}}}{L_{\mathrm{d}}} & 0 \\ 0 & \dfrac{T_{\mathrm{s}}}{L_{\mathrm{q}}} \end{bmatrix}, \qquad (1)$$

where R_{s} is the stator resistance, T_{s} represents the sampling period, ω_{e} is the electric angular speed and L_{d} and L_{q} are the dq inductances. $\boldsymbol{h}(k) = [0,\, \omega_{\mathrm{e}}\Lambda_{\mathrm{pm}}]^{\mathrm{T}}$ is the back electro-motive force induced by the permanent magnet flux linkage Λ_{pm}. The input of the plant are the dq voltages $\boldsymbol{u}_{\mathrm{dq}}$, while the state variables are the dq current $\boldsymbol{i}_{\mathrm{dq}}$.

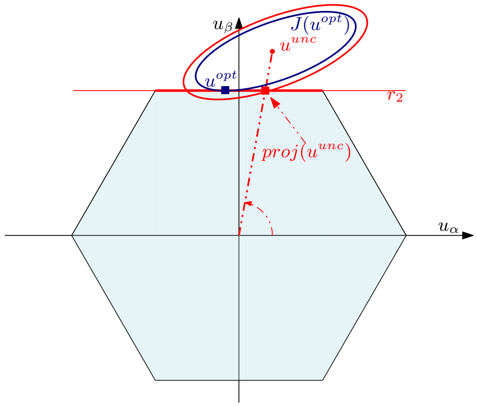


Fig. 1: Difference between constrained solution of the control problem and a phase-preserving projection (*proj*) of the unconstrained solution. Ellipses represent the cost function level sets.

B. The Data-Enabled predictive control problem

The DeePC is a purely data-driven control technique, whose design consists of two main steps. First, a data collection experiment is conduced on the motor under test. An input voltage excitation is applied to the motor windings. The excitation voltage signal and the resulting stator currents are stored. We refer to [7] for more details about the data collection stage, since it is out of the scope of this work.

Once data are available, the desired control problem needs to be defined. The distinguishing feature of DeePC is the overcoming of system identification procedures on the data-set. Raw data appears reshaped in the form of Hankel matrices. If $\boldsymbol{y}_l$ is a l-long vector of currents samples, the related Hankel matrix is built:

$$\mathscr{H}(\boldsymbol{y}_d) := \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_{l-T_{\mathrm{ini}}-N+1} \\ \boldsymbol{y}_2 & \boldsymbol{y}_3 & \cdots & \boldsymbol{y}_{l-T_{\mathrm{ini}}-N+2} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{y}_{T_{\mathrm{ini}}+N} & \boldsymbol{y}_{T_{\mathrm{ini}}+N+1} & \cdots & \boldsymbol{y}_l \end{bmatrix}, \qquad (2)$$

where T_{ini} and N are two tuning parameters, accounting for lower bounds of the lag and the order of the unknown model, respectively. Incremental Hankel matrices are built to add an integral action to the controller, as in [7]. In particular, such integral action rejects all the speed-dependent time-variant disturbances that characterize the PMSM currents dynamic (1).

The online problem is defined accordingly to the MPC framework. Considering the current control of PMSM, the program consists into optimizing the sequence of future voltages $\boldsymbol{u} = [\boldsymbol{u}_{\mathrm{dq}}(k),\, \boldsymbol{u}_{\mathrm{dq}}(k+1),\, \ldots,\, \boldsymbol{u}_{\mathrm{dq}}(k+N-1)]^{\mathrm{T}}$ such that the future currents $\boldsymbol{y} = [\boldsymbol{i}_{\mathrm{dq}}(k+1),\, \boldsymbol{i}_{\mathrm{dq}}(k+2),\, \ldots,\, \boldsymbol{i}_{\mathrm{dq}}(k+N)]^{\mathrm{T}}$ are steered to a desired reference $\boldsymbol{r} = [\boldsymbol{i}^*_{\mathrm{dq}}(k+1),\, \boldsymbol{i}^*_{\mathrm{dq}}(k+

2), ..., $\boldsymbol{i}^*_{\mathrm{dq}}(k+N)]^{\mathrm{T}}$. Thus, the optimization program is:

$$\min_{\boldsymbol{g},\boldsymbol{u},\boldsymbol{y}} \quad ||\boldsymbol{y}-\boldsymbol{r}||^2_{\mathbf{Q}} + ||\boldsymbol{u}||^2_{\mathbf{R}} + \lambda_g||\boldsymbol{g}||^2 \tag{3a}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{U}_{\mathrm{P}} \\ \mathbf{Y}_{\mathrm{P}} \\ \mathbf{U}_{\mathrm{F}} \\ \mathbf{Y}_{\mathrm{F}} \end{bmatrix} g = \begin{bmatrix} \boldsymbol{u}_{\mathrm{ini}} \\ \boldsymbol{y}_{\mathrm{ini}} \\ \boldsymbol{u} \\ \boldsymbol{y} \end{bmatrix}, \tag{3b}$$

$$\boldsymbol{u}_{\mathrm{dq}}(k) \in \mathcal{U},\ k = 0, ..., N-1. \tag{3c}$$

$\mathbf{U}_{\mathrm{P}}$, $\mathbf{Y}_{\mathrm{P}}$, $\mathbf{U}_{\mathrm{F}}$ and $\mathbf{Y}_{\mathrm{F}}$ are block-Hankel matrices (see definition in Sec. II of [18]) of the pre-collected voltage/current data. The vectors $\boldsymbol{u}_{\mathrm{ini}}$ and $\boldsymbol{y}_{\mathrm{ini}}$ are the dq voltages and dq currents, respectively, sampled the previous control period. Moreover, the vector $\boldsymbol{g} \in \mathcal{R}^{l\times 1}$ is the auxiliary variable for which the optimization needs to be solved. A regularization on $\boldsymbol{g}$ is included in the problem by the weight λ_g. Q and R are two additional weighting matrices. Finally, $\mathcal{U}$ is the feasible set for the input voltage. The set has an hexagonal shape for PMSMs fed by two-levels inverters [14], which is the topology considered hereinafter in this work. Nevertheless, the same controller can be implemented on other power converters topologies, provided that the feasible voltage set is defined accordingly.

After some manipulations, problem (3) is rearranged as a standard QP in the unknown g:

$$\min_{\boldsymbol{g}} \quad \frac{1}{2}\boldsymbol{g}^{\mathrm{T}}\mathbf{H}\boldsymbol{g} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{g} \tag{4a}$$

$$\text{subject to} \quad \mathbf{A}_{\mathrm{eq}}\boldsymbol{g} = \boldsymbol{b}_{\mathrm{eq}} \tag{4b}$$

$$\mathbf{A}_{\mathrm{in}}\boldsymbol{g} \leq \boldsymbol{b}_{\mathrm{in}} \tag{4c}$$

where $\mathbf{H} = 2\big(\mathbf{Y}_{\mathrm{F}}^{\mathrm{T}}\mathbf{Q}\mathbf{Y}_{\mathrm{F}} + \mathbf{U}_{\mathrm{F}}^{\mathrm{T}}\mathbf{R}\mathbf{U}_{\mathrm{F}} + \lambda_g\mathbf{I}_d\big)$, $\boldsymbol{c} = 2\mathbf{Y}_{\mathrm{F}}^{T}\mathbf{Q}\boldsymbol{r}$, $\mathbf{A}_{\mathrm{eq}} = [\mathbf{U}_{\mathrm{P}}, \mathbf{Y}_{\mathrm{P}}]^{\mathrm{T}}$, $\boldsymbol{b}_{\mathrm{eq}} = [\boldsymbol{u}_{\mathrm{ini}}, \boldsymbol{y}_{\mathrm{ini}}]^{\mathrm{T}}$, $\mathbf{A}_{\mathrm{in}} = \mathbf{M}\mathbf{T}_{\mathrm{P}}^{-1}\mathbf{U}_{\mathrm{F}}$ and $\boldsymbol{b}_{\mathrm{in}} = 2u_{\mathrm{DC}}/\sqrt{3}\,[1, 0.5, 1, 1, 0.5, 1]^{\mathrm{T}}$. Moreover, u_{DC} is the inverter DC bus voltage, $\mathbf{T}_{\mathrm{P}}$ is the Park transformation matrix, $\mathbf{I}_l$ is the l-wide identity matrix and

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ \sqrt{3} & 0 & -\sqrt{3} & -\sqrt{3} & 0 & \sqrt{3} \end{bmatrix}^{\mathrm{T}}.$$

The latter describes the hexagon voltage constraint as in [19]. Being a physic limitation of the adopted power converter topology, the hexagonal constraints hold at any point in time. Among all the just defined matrices and vectors, $\mathbf{H}$ and $\mathbf{A}_{\mathrm{eq}}$ are constant in the quadratic program (QP), thus they can be pre-processed offline. On the opposite, $\boldsymbol{c}$, $\boldsymbol{b}_{\mathrm{eq}}$, $\mathbf{A}_{\mathrm{in}}$ and $\boldsymbol{b}_{\mathrm{in}}$ are influenced by new measured streaming data and they need to be partially recomputed each control period.

The unconstrained solution $\boldsymbol{u}^{\mathrm{unc}}$ of (4), i.e., the solution neglecting (4c), was previously presented in [7]. Infeasible input voltages were projected within the feasible set preserving the voltage vector phase, as shown in Fig. 1. Other projections strategies may be considered, i.e. orthogonal projections on the constraints. The projected solution may represent a good approximation of the optimum in some conditions. However, the same figure highlights the relevance of finding the constrained solution of the problem. In fact, the optimal solution $\boldsymbol{u}^{\mathrm{opt}}$ which minimizes the cost (3) could be far from the projected unconstrained solution, depending on the contour of the cost level-sets [14].

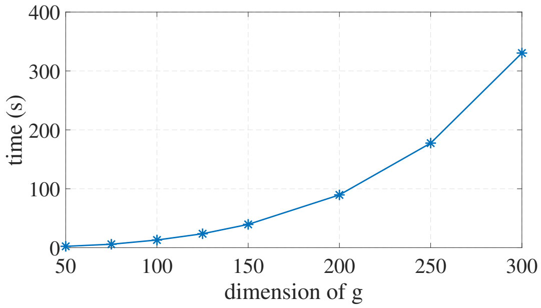


Fig. 2: Time for solving 1000 DeePC instances versus length of $\boldsymbol{g}$.

A constrained form of the DeePC is hard to implement in embedded applications, such as the control of PMSM currents. In fact, the computation time bottleneck is tight, while the problem size blows up quickly when increasing the amount of available data. Fig. 2 represents the time required to solve 1000 instances of the DeePC constrained problem using qpOASES [20] as solver, while increasing the dimension of $\boldsymbol{g}$, which depends on the amount of data pre-collected for building the DeePC (Sec. III. A of [7]). The specific scenario giving rise to Fig.1 is described in Section IV. The computational complexity grows as $\mathcal{O}(d^3)$. Thus, a different approach is proposed to unlock the real-time feasibility of constrained DeePC.

III. Real-time feasible data-driven control

In order to enable the real-time feasibility of the constrained DeePC, this paper proposes the implementation of a customized solver for the online optimization and a problem reduction method based on the POD. Fig. 3 resumes the proposed methodology. Once input/output voltage/current samples are collected from the PMSM (Section III A [7]), the DeePC optimization problem is formulated. Then, such a problem is reduced by means of a POD. The problem reduction is described in Section III-B, distinguishing between the constrained and unconstrained case. Finally, the reduced problem is solved online in real-time by using the customized solver described in Section III-A. The adopted solver is described first, to ease the comprehension of the interaction between itself and the DeePC problem reduction.

A. Solver for the constrained DeePC problem

The optimization behind the constrained DeePC is solved by using a customized version of the method proposed in [14]. The solver is briefly described in order to highlight where the main improvements are added.

First, the solver computes the unconstrained solution, $\boldsymbol{g}^{\mathrm{unc}}$, of (4a) including the equality constraints (4b). However, this computation presents troubles related to the fact that the dimension of $\boldsymbol{g}$ depends on the amount of data stored in the Hankel matrix $\mathbf{A}_{\mathrm{eq}}$. Sec. III-B proposes a problem reduction technique to handle this issue.

Then, the solver performs a feasibility check of the input solution $\boldsymbol{u}^{\mathrm{unc}} = \mathbf{U}_{\mathrm{F}}\boldsymbol{g}^{\mathrm{unc}}$. Since the feasible voltage set is symmetric Fig.1, four possible cases can occur:

- the unconstrained solution is feasible;

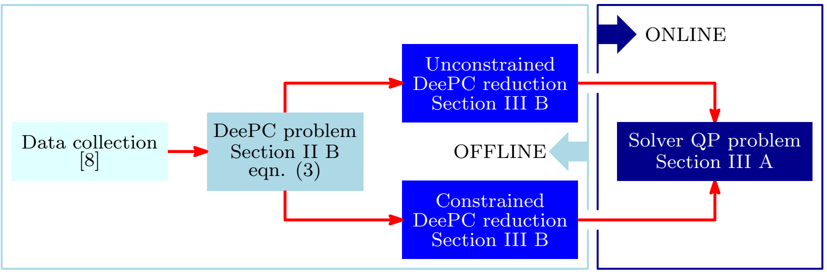

ONLINE
Data collection [8]
DeePC problem Section II B eqn. (3)
Unconstrained DeePC reduction Section III B
OFFLINE
Constrained DeePC reduction Section III B
Solver QP problem Section III A

Fig. 3: Structure diagram of the proposed methodology.

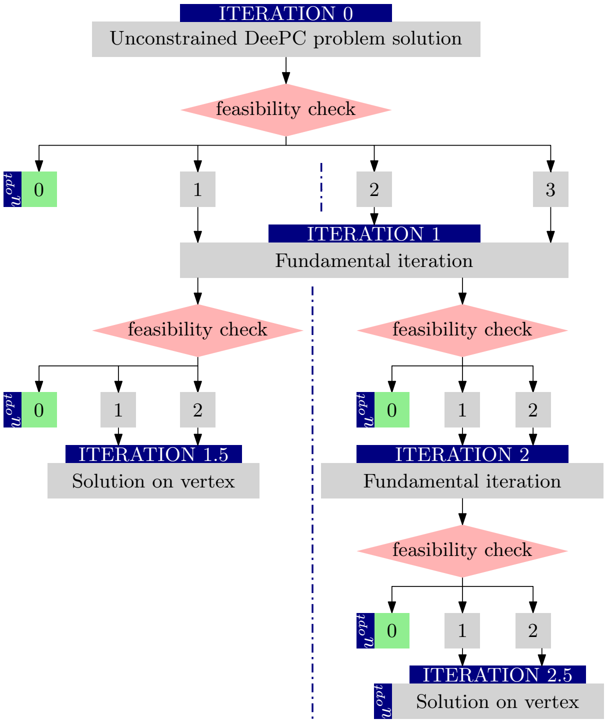


Fig. 4: Flow-chart of the adopted QP solver.

- $\boldsymbol{u}^{\text{unc}}$ violates one constraint of the hexagon;
- $\boldsymbol{u}^{\text{unc}}$ violates two consecutive constraints of the hexagon;
- $\boldsymbol{u}^{\text{unc}}$ violates three consecutive constraints of the hexagon.

In the first case, the DeePC optimum coincides with the unconstrained solution (*iteration 0* in Fig. 4). In all the other cases typically encountered in practice, the solver needs to compute at most two *fundamental iterations*. A bounded number of iterations is the key features of the adopted solver and it is described in [21]. In a nutshell, this feature is mainly related to the fact the selected control horizon for the DeePC is equal to one, regardless the prediction horizon length.

In case of unfeasible unconstrained solutions, the solver proceeds considering one violated constraint at a time, penalizing such constraint in the cost function. Assuming that the j-th constraint is violated, the solver searches the best solution lying on the jth line. Since the optimization variable is still $\boldsymbol{g}$, the same issue of the unconstrained problem occurs. Sec. III-B describes the proposed reduction method adopted to speed up the *fundamental iteration*.

The new solution may be feasible or not again, thus a new feasibility check is performed. If the solution is feasible, then it is optimal also for (4). This is the case shown in Fig. 1. $\boldsymbol{u}^{\text{unc}}$ (associated to $\boldsymbol{g}^{\text{unc}}$) violates the line r_2, while the solution computed on r_2 lies on the hexagon segment, thus it is feasible and optimal. Depending on the number of violated constraints, this iteration is repeated or the solution is searched among the hexagon vertices. Thanks to the geometry of the problem, no more than two *fundamental iterations* are needed [14]. Fig. 4 provides a picture of all the iterations that could occur. The feasibility checks on the solutions are denoted by reddish blocks. Greenish blocks indicate the nodes where the solution computed on a segment is feasible. Thus, the solver stops exploring the decision tree. On the contrary, grayish blocks denote nodes where there are still violated hexagon constraints.

B. Control Problem Reduction

Unconstrained DeePC — The equality constrained DeePC is obtained by neglecting all the inequality constraints in (4). The Karush-Kuhn-Tucker (KKT) system associated to the problem is

$$\begin{bmatrix}\mathbf{H} & \mathbf{A}_{\text{eq}}^{\mathrm{T}}\\ \mathbf{A}_{\text{eq}} & 0\end{bmatrix}\begin{pmatrix}\boldsymbol{g}\\ \boldsymbol{\mu}_{\text{eq}}\end{pmatrix} = \begin{pmatrix}-\boldsymbol{c}\\ \boldsymbol{b}_{\text{eq}}\end{pmatrix} \rightarrow \mathbf{K}_{\text{eq}}\gamma = \boldsymbol{\rho}. \tag{5}$$

Inspecting the expressions of the Hessian and the linear term of the QP, the solution of the KKT system can be parametrized in an affine manner as:

$$\mathbf{K}_{\text{eq}}\,\gamma = \mathbf{c}_1\,\boldsymbol{r} + \mathbf{c}_2\begin{bmatrix}\boldsymbol{u}_{\text{ini}}\\ \boldsymbol{y}_{\text{ini}}\end{bmatrix}, \tag{6}$$

where $\boldsymbol{r}, \boldsymbol{u}_{\text{ini}}$, and $\boldsymbol{y}_{\text{ini}}$ are considered as parameters of the control problem. It is worth noting that matrix $\mathbf{K}_{\text{eq}}$ has dimensions that depend on the number of collected data that can be arbitrarily large (e.g., even more than some hundreds). Thus, solving (6) is prohibitive for real-time applications.

The reduction of the unconstrained DeePC is achieved by applying a parametric POD [16]. The algorithm spans the solution space in order to construct a proper basis, and then it implements the Galerkin projection to find out a reduced order control problem. In particular, the auxiliary variable $\hat{\gamma}$ is introduced, and the reduction algorithm (described in Algorithm 1 [22]) iteratively constructs the projection matrix

$$\mathbf{V} = \begin{bmatrix}\mathbf{V}_g\\ \mathbf{V}_\mu\end{bmatrix},$$

such that $\tilde{\boldsymbol{g}} = \mathbf{V}_g\hat{\gamma}$ and $\tilde{\boldsymbol{\mu}}_{eq} = \mathbf{V}_\mu\hat{\gamma}$, i.e., $\tilde{\gamma} = \mathbf{V}\hat{\gamma}$, where the symbol ˜ indicates approximated quantities. For the specific application discussed in this paper, the size of $\hat{\boldsymbol{\gamma}}$ is much smaller than the size of $\boldsymbol{\gamma}$ (i.e., the number of columns of $\mathbf{V}$ is much less than its rows). Thus, by projecting the original full control problem by using $\mathbf{V}$, the size (and therefore the computational cost) of the problem is significantly reduced.

Therefore, once $\mathbf{V}$ is constructed, the reduced control problem can be obtained by applying a Galerkin projection of (6) by using $\mathbf{V}$, i.e.,

$$\hat{\mathbf{K}}_{\text{eq}} = \mathbf{V}^{\text{T}}\mathbf{K}_{\text{eq}}\mathbf{V}, \tag{7}$$

$$\hat{\mathbf{c}}_1 = \mathbf{V}^{\text{T}}\mathbf{c}_1, \tag{8}$$

$$\hat{\mathbf{c}}_2 = \mathbf{V}^{\text{T}}\mathbf{c}_2, \tag{9}$$

thus leading to

$$\hat{\mathbf{K}}_{\text{eq}}\,\hat{\boldsymbol{\gamma}} = \hat{\mathbf{c}}_1\,\boldsymbol{r} + \hat{\mathbf{c}}_2\,[\boldsymbol{u}_{\text{ini}},\,\boldsymbol{y}_{\text{ini}}]^{\text{T}}. \tag{10}$$

The solution of the reduced control problem is achieved by solving (10) and then an approximate solution $\tilde{\boldsymbol{g}}$ of the full order control problem is obtained as $\tilde{\boldsymbol{g}} = \mathbf{V}_g\hat{\boldsymbol{g}}$.

It is desirable that the approximate solution of the KKT system grants a minimal relative error

$$\epsilon = \frac{|\boldsymbol{g} - \tilde{\boldsymbol{g}}|}{|\boldsymbol{g}|} = \frac{|\boldsymbol{g} - \mathbf{V}_g\hat{\boldsymbol{\gamma}}|}{|\boldsymbol{g}|} \tag{11}$$

with respect to the solution of the original full order problem (5). It is worth noting that, the error ϵ is evaluated only with respect to $\boldsymbol{g}$, while the accuracy on the re-construction of $\boldsymbol{\mu}_{eq}$ is ignored. Indeed, the optimal input voltage $\boldsymbol{u}^{opt}$ (which is the final quantity to compute) only depends on $\boldsymbol{g}$ as $\boldsymbol{u}_{opt} = \mathbf{U}_{\text{F}}\boldsymbol{g}$. The choice of considering the error ϵ as in (11) and not with respect to the solution $\boldsymbol{\gamma}$ (i.e., including also $\boldsymbol{\mu}_{eq}$) allows for obtaining a small and well conditioned reduced control problem.

For this reason, the construction of $\mathbf{V}$ by means of Algorithm 1 is performed iteratively by adopting a pseudo-random POD method [23]. At each iteration of Algorithm 1, the tentative basis $\mathbf{V}$ is used to reduce the dimension of the problem (i.e., constructing (6)) and the accuracy of the reduced problem with respect to the full order one is tested against several random choices of parameters $\boldsymbol{r}, \boldsymbol{u}_{\text{ini}}$, and $\boldsymbol{y}_{\text{ini}}$. If for each selection of the random parameters the error ϵ is lower than the required tolerance η, then the Algorithm stops. Otherwise, the parameter selection which gave the maximum error is selected and the algorithm performs a new iteration by solving the full order problem with respect to this parameter selection and the related solution is added to the basis $\mathbf{V}$ and then orthogonalized, i.e.,

$$\mathbf{V} \leftarrow \text{GSO}([\mathbf{V}, \boldsymbol{\gamma}]). \tag{12}$$

where GSO is the Gram-Schmidt orthogonalization. It is worth noting that the equation above should be intended in an algorithmic sense, i.e. the new projection matrix $\mathbf{V}$ (left-hand-side of (12)) is updated by adding to the old $\mathbf{V}$ (right-hand-side of (12)) a new column (i.e. $\boldsymbol{\gamma}$) and applying the GSO to the resulting matrix. At iteration one of the Algorithm, since $\mathbf{V}$ has not been created yet, $\mathbf{V}$ at the right-hand-side of (12) is obviously taken as the empty matrix. Other stopping criteria may be used when generating the ortho-normal basis $\mathbf{V}$. For instance, bounds on the problem dimension can be imposed in case of more complicated applications. In this work we adopted a stopping criterion based on the problem reduction error, which is by far the most widespread.

Algorithm 1 Unconstrained problem reduction

```
Input: Matrices $\mathbf{K}_{\text{eq}}$, $\boldsymbol{b}_{\text{eq}}$ and $\boldsymbol{c}$, parametric in $\boldsymbol{u}_{\text{ini}}, \boldsymbol{y}_{\text{ini}}$ and $\boldsymbol{r}$. The parameters ranges are $[u_{\min}, u_{\max}]$, $[y_{\min}, y_{\max}]$, $[r_{\min}, r_{\max}]$ for $\boldsymbol{u}_{\text{ini}}$, $\boldsymbol{y}_{\text{ini}}$, and $\boldsymbol{r}$ elements, respectively.
Step $\{0\}$ Set initial values for parameters, for instance the mean values of all the parameters, i.e., $\frac{u_{\min}+u_{\max}}{2}$, $\frac{y_{\min}+y_{\max}}{2}$, $\frac{r_{\min}+r_{\max}}{2}$
Set $\epsilon_\star = +\infty$
Set $N_{\text{rand}}$ (e.g. $N_{\text{rand}} = 20$)
Set a desired value of $\eta$ (e.g. $\eta = 10^{-4}$)
while $\epsilon_\star > \eta$ do
    Step $\{1\}$ Find the solution $\boldsymbol{\gamma} = [\boldsymbol{g},\, \boldsymbol{\mu}_e]^{\text{T}}$ of (6)
    Step $\{2\}$ Update the orthonormal basis $\mathbf{V}$, i.e. (12)
    Step $\{3\}$ Generate/Update the reduced order problem, i.e., (7)-(9)
    Step $\{4\}$ Generate $N_{\text{rand}}$ random values of $\boldsymbol{u}_{\text{ini}}$, $\boldsymbol{y}_{\text{ini}}$ and $\boldsymbol{r}$
    for $h = 1, \cdots, N_{\text{rand}}$ do
        Select the $h$-th random set of parameters,
        Step $\{5\}$ Find the solution $[\hat{\boldsymbol{g}},\, \hat{\boldsymbol{\mu}}_e]^{\text{T}}$ of the reduced order problem (10)
        Step $\{6\}$ Evaluate the residual $\epsilon$ (11), with respect to the full order problem
    end for
    Step $\{7\}$ Find the set of parameters generated at step 4 which maximizes the residual and assign the corresponding maximum residual to $\epsilon_\star$. The new parameter set is used for solving step 1
end while
Output: Reduced order model and the projection matrix, i.e., $\hat{\mathbf{K}}_{eq} = \mathbf{V}^{\text{T}}\mathbf{K}_{eq}\mathbf{V}$;  $\hat{\mathbf{c}}_1 = \mathbf{V}^{\text{T}}\mathbf{c}_1$;  $\hat{\mathbf{c}}_2 = \mathbf{V}^{\text{T}}\mathbf{c}_2$
```

Finally, the online controller implements only a simple linear feedback rule to compute the approximate optimal input voltage $\tilde{\boldsymbol{u}}^{\text{opt}}$:

$$\tilde{\boldsymbol{u}}^{\text{opt}} = \mathbf{U}_{\text{F}}\mathbf{V}_g\hat{\mathbf{K}}_{\text{eq}}^{-1}\left(\hat{\mathbf{c}}_1\boldsymbol{r} + \hat{\mathbf{c}}_2\begin{bmatrix}\boldsymbol{u}_{\text{ini}}\\ \boldsymbol{y}_{\text{ini}}\end{bmatrix}\right), \tag{13}$$

where $\mathbf{U}_{\text{F}}\mathbf{V}_g\hat{\mathbf{K}}_{\text{eq}}^{-1}\hat{\mathbf{c}}_1$ and $\mathbf{U}_{\text{F}}\mathbf{V}_g\hat{\mathbf{K}}_{\text{eq}}^{-1}\hat{\mathbf{c}}_2$ are both computed offline. Results will show that the accuracy of the approximated solution $\tilde{\boldsymbol{u}}^{\text{opt}}$ is actually very high and almost identical to the exact one $\boldsymbol{u}^{\text{opt}}$.

Constrained DeePC — When the unconstrained solution is not feasible, the solver performs at most two *fundamental iterations*, as in [14]. The reduction method needs to be readjusted for these iterations, since the parametrization (6) is no more usable.

Recall that the solver consider always one constraint at a time, penalizing that constraint in the cost function. The methodology adopted to reduce the constrained program is

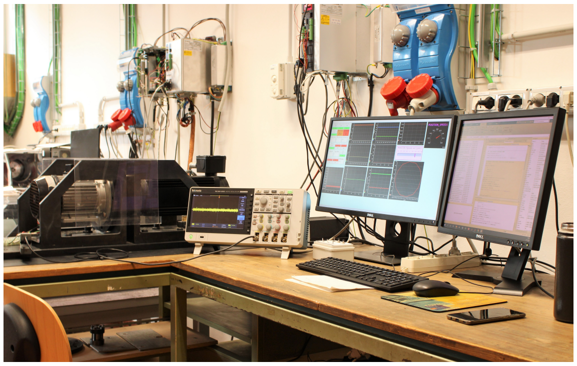


Fig. 5: Test-bed layout.

similar to the one adopted for the unconstrained one. First, the KKT system associated to the j-th constraint is introduced:

$$\begin{bmatrix} \mathbf{H} & \mathbf{A}_{\text{eq}}^{\mathrm{T}} & \mathbf{A}_{\text{in}}(j,:)^{\mathrm{T}} \\ \mathbf{A}_{\text{eq}} & 0 & 0 \\ \mathbf{A}_{\text{in}}(j,:) & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{\mu}_{\text{eq}} \\ \boldsymbol{\mu}_{\text{in}} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{c} \\ \boldsymbol{b}_{\text{eq}} \\ \boldsymbol{b}_{\text{in (j)}} \end{pmatrix}. \quad (14)$$

It is worth mentioning that the parametrization of this system is much richer than the one of the unconstrained program. In fact, $\mathbf{A}_{\text{in}}$ encapsulates the Park transformation, i.e., the rotor position information and the DC bus voltage magnitude. Furthermore, with the proposed solver, the violated constraint j becomes an additional parameter of the problem.

Then, the KKT system is rewritten, highlighting the affine relationships with parameters. In particular, the KKT inequality matrix $\mathbf{K}_{in}$ is parametrized as

$$\begin{aligned} \mathbf{K}_{in} =& \mathbf{H}_1 + \mathbf{H}_2\mathbf{M}(j,1)\cos\theta_e + \mathbf{H}_3\mathbf{M}(j,2)\cos\theta_e \\ &+ \mathbf{H}_4\mathbf{M}(j,1)\sin\theta_e + \mathbf{H}_5\mathbf{M}(j,2)\sin\theta_e, \end{aligned} \quad (15)$$

where θ_e is the electric motor position, while the KKT right-hand side $\boldsymbol{\rho}_{\text{in}}$ is parametrized as

$$\boldsymbol{\rho}_{\text{in}} = \mathbf{c}_3\,\boldsymbol{r} + \mathbf{c}_3\left[\boldsymbol{u}_{\text{ini}},\ \boldsymbol{y}_{\text{ini}}\right]^{\mathrm{T}} + \mathbf{c}_5\,u_{\text{DC}}. \quad (16)$$

As for the unconstrained DeePC, the problem is reduced using the same random POD algorithm, whose outputs are the projection matrix $\mathbf{V}_{\text{in}}$, $\hat{\mathbf{K}}_{\text{in}}(\theta_e, j) = \mathbf{V}_{\text{in}}^{\mathrm{T}}\mathbf{K}_{\text{in}}\mathbf{V}_{\text{in}}$, $\hat{\mathbf{c}}_3 = \mathbf{V}_{\text{in}}^{\mathrm{T}}\mathbf{c}_3$, $\hat{\mathbf{c}}_4 = \mathbf{V}_{\text{in}}^{\mathrm{T}}\mathbf{c}_4$, and $\hat{\mathbf{c}}_5 = \mathbf{V}_{\text{in}}^{\mathrm{T}}\mathbf{c}_5$. The only difference between the reduction techniques are due to an enlarged set of parameters. Thus, the optimal solution of (4) lying on the violated constraint j is

$$\boldsymbol{u}_r^{\mathrm{j}} = \hat{\mathbf{K}}_{\text{in}}(\theta_e, j)^{-1}\left(\hat{\mathbf{c}}_3\boldsymbol{r} + \hat{\mathbf{c}_4}[\boldsymbol{u}_{\text{ini}},\ \boldsymbol{y}_{\text{ini}}]^{\mathrm{T}} + \hat{\mathbf{c}}_5 u_{\text{DC}}\right). \quad (17)$$

This computation is repeated at most two times when implementing the selected QP solver.

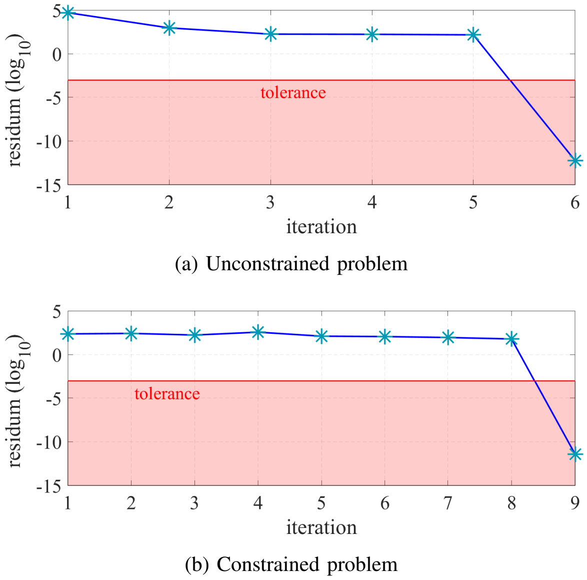
IV. Results

The proposed constrained DeePC algorithm has been validated by means of both simulations and experiments.

On one hand, all the simulations were run on an Intel(R) Core(TM) i7-8700 CPU 3.20GHz. On the other hand, experimental tests were performed by using the test-bench whose layout is reported in Fig. 5. In particular, a back-to-back configuration is adopted. The load motor is controlled in speed control mode, whereas the tested motor is operated in current control mode. An interior permanent magnet motor is considered for the validation, whose parameters are resumed in Table I. Control algorithms are real-time implemented on a dSPACE MicroLabBox platform, at a sample rate of $T_s = 100\ \mu s$, in accordance with the nominal switching frequency of the inverter. The data-driven control finds the optimal voltage references feeding the motor. Then, a space vector modulation algorithm computes the duty-cycles for the inverter switches. Finally, switching pulses are generated by means of the dSPACE libraries. All the reported currents are normalized with respect to the nominal value I_N, while the voltages are normalized with respect to the DC bus voltage U_{DC}, both specified in Table I.

TABLE I: Overview of the drive parameters.

Parameter	Symbol	Value
Pole pairs	p	3
Phase resistance	R_s	$1\,\Omega$
d-axis inductance	L_d	0.010 H
q-axis inductance	L_q	0.014 H
PM flux-linkage	Λ_{pm}	0.26 V s
Nominal current	I_N	6.2 Arms
Nominal d current	$I_{N,d}$	-1.1 A
Nominal q current	$I_{N,q}$	8.7 A
Nominal speed	Ω_N	1000 rpm
DC bus voltage	U_{DC}	200 V
Sampling period	T_s	$100\ \mu s$



(a) Unconstrained problem

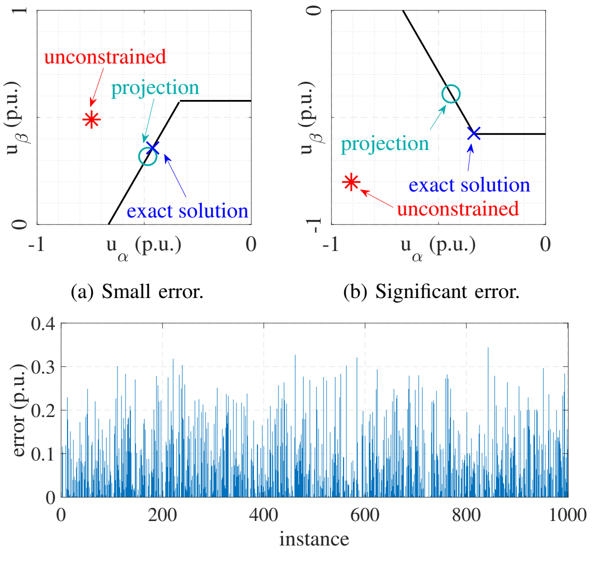
(b) Constrained problem

Fig. 6: Typical convergence curve of the QP problem reduction.

A. Problem reduction

The dimensionality reduction of the DeePC problem is here analyzed in a MATLAB Simulink environment. A set of 103 input/output voltage/current samples was collected form the IPM machine, following the guidelines described in [7]. In particular, the data-collection stage was performed at standstill with a magnitude of the excitation voltage signal equal to the 10% of the bus DC voltage.

Following the data-driven design procedure [7], the DeePC horizon lengths T_{ini} and N were set respectively to 1 and 3.



(a) Small error.

(b) Significant error.

(c) Error between the solution considering 1000 instances.

Fig. 7: Comparison between optimal solution of the QP problem and the projected unconstrained solution. Voltages are normalized with respect to U_{DC}.

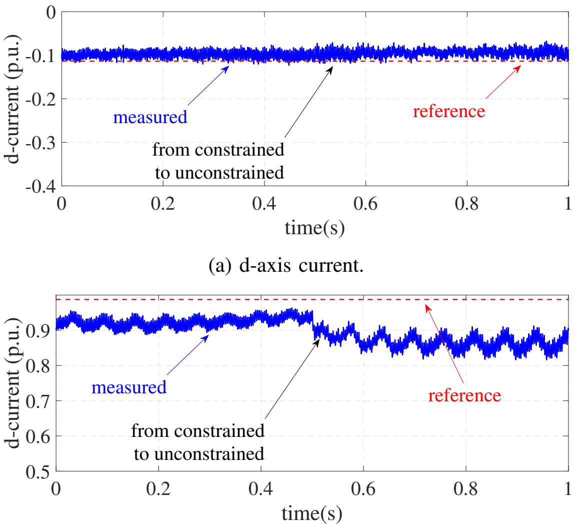
Since 103 samples were taken, the dimension of the decision variable $\boldsymbol{g}$, i.e., d, results to be $d = T - T_{\text{ini}} - N + 1 = 103 - 1 - 3 + 1 = 100$. However, this problem size is unaffordable for real-time applications, especially at high sampling rating and with limited computation resources.

The problem reduction algorithm (Algorithm 1) is run on the data-set to compress both the unconstrained and constrained QP problem. Fig. 6 shows the convergence curve of the POD method, given the selected tolerance $\eta = 10^{-3}$. Both the constrained and the unconstrained programs are reduced of a factor higher than one order of magnitude. In fact, it is reminded that $\boldsymbol{g}$ is a 100-long vector. In addition, the dimension of the reduced problem is not influenced by the dimension of the data-set. In fact, the POD converges with equal iterations even if longer snapshots are elaborated. This feature is coherent with the fact that the dimension of the problem should be determined by the DeePC design, i.e., the value of N, T_{ini}, and the considered plant. Indeed, this method acts as a rank revealing technique of the problem without making any a-priori assumptions.

B. Difference between constrained and unconstrained DeePC

A common practice introduced by engineers when implementing MPC controllers in the power electronic area is the use of the unconstrained version to limit the computation burden. However, this practice leads to the application of sub-optimal input voltages to the motor. This consideration is valid for all the predictive controllers, including the DeePC.

Difference between constrained and constrained solutions of a QP problem occurs when the unconstrained solution is not feasible, as shown in Fig. 7. In fact, infeasible solutions need to be handled somehow, when the DeePC is applied in the unconstrained fashion. A common practice consists into projecting unconstrained infeasible solutions on the hexagonal feasible voltage domain. However, such solution may be far from the true optimum. The latter is computed by the constrained DeePC by searching the optimal solutions lying on the boundaries.



(a) d-axis current.

(b) q-axis current.

Fig. 8: Current tracking performance at the boundaries of the feasible voltage set.

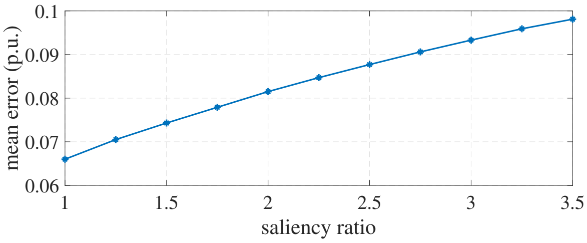
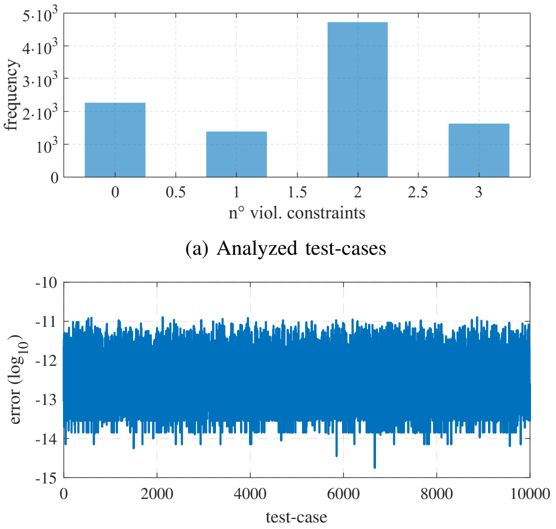


Fig. 9: Mean error committed by the approximated solutions with respect to the exact ones as function of the motor saliency, i.e. $L_{\text{q}}/L_{\text{d}}$.

The error between the two solutions depends on the operating point. Fig. 7(a) and Fig. 7(b) illustrate two significant cases characterized by a small and large discrepancy, respectively. It is relevant that the Euclidean distance between true optima and the projected solutions can be larger than 30% of the DC bus voltage, as observed by running several instances (Fig. 7(c)). The benefits achieved by implementing the constrained solution are experimentally validated, too. The motor under test was controlled in current mode, while the load machine imposed an operating speed equal to the nominal one. The inverter DC link voltage was reduced of a 10% factor to let the DeePC controller compute often voltages on the hexagonal set boundaries. Results are shown in Fig.8 in terms of dq currents. In the first half of the test the constrained DeePC solutions fed the motor, while the projected unconstrained ones in the second half. Both algorithms did no allow to track the desired current reference, since the inverter ran out of voltage. However, the constrained solutions a lower bias error and lower oscillations with respect to the unconstrained ones. Thus, the constrained controller exploits more effectively the available voltage set.

A second parameter that influence the accuracy of the



(a) Analyzed test-cases

(b) Reduced problem error

Fig. 10: Relative error between the QP problem solution and reduced approximated QP problem.

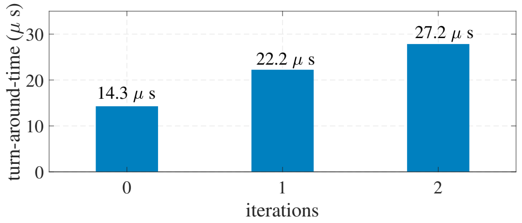


Fig. 11: Turnaround time of the online program as function of the number of iterations (Fig. 4).

projected solution is the saliency of the adopted PMSM, i.e., the $L_\mathrm{q}/L_\mathrm{d}$ ratio. The set of simulations performed to build Fig.7 are repeated for several values of L_q, keeping constant L_d. A reduction of the motor saliency is experienced by interior permanent magnet motors in presence of iron-saturation, which implies in turns a reduction of L_q. The mean error between projected and optimal solutions is evaluated and reported in Fig. 9. The mean error grows clearly with the saliency. This is due to the fact that the DeePC cost function (3) contour lines of Fig. 1 become more and more elliptical and the projection technique becomes more and more inaccurate. This observation suggests that the implementation of the constrained DeePC program results of particular relevance for highly anisotropic machines.

C. Accuracy of the approximated DeePC problem solution

As a further step in the validation, the controller matrices are plugged in the solver (Fig.4). The most relevant feature desired from a problem reduction routine is that the solutions of the approximated control problem and the ones of the large original problem are similar. For this reason, thousand of DeePC problem instances (4) were launched. The DeePC instances are generated randomly choosing the PMSM current references $\boldsymbol{r}$, the initial voltage and current conditions $\boldsymbol{u}_\mathrm{ini}$, $\boldsymbol{y}_\mathrm{ini}$ and the electric motor position and speed. The number of instances are selected in order to cover all the possible solver cases, previously resumed in Fig.4. Fig. 10 resumes the accuracy of the approximated problem solution. In particular, Fig. 10(a) confirms that the analyzed instances cover all the possible cases in terms of number of violated constraints.

The large original DeePC problem is solved by means of qpOASES, an efficient QP solver proposed by *Ferrau et al.* [24]. The comparison of the euclidean distance between the auxiliary variables could be used as an alternative. Fig. 10(b) shows that the error committed when solving the reduced problem is practically null for all the instances.

D. Real-time feasibility

The proposed constrained DeePC is then implemented in real-time on a dSPACE MicroLabBox, which is equipped with a 2 GHz NXP QorIQ P5020 microprocessor. The motor under test is dragged at $110\%\,\Omega_\mathrm{N}$ by the load machine. In the while, the motor under test is controlled in current mode and the nominal currents are set as reference. This allows the solver to find many infeasible working points and, ultimately, to explore all the case of infeasibility presented in Fig. 4.

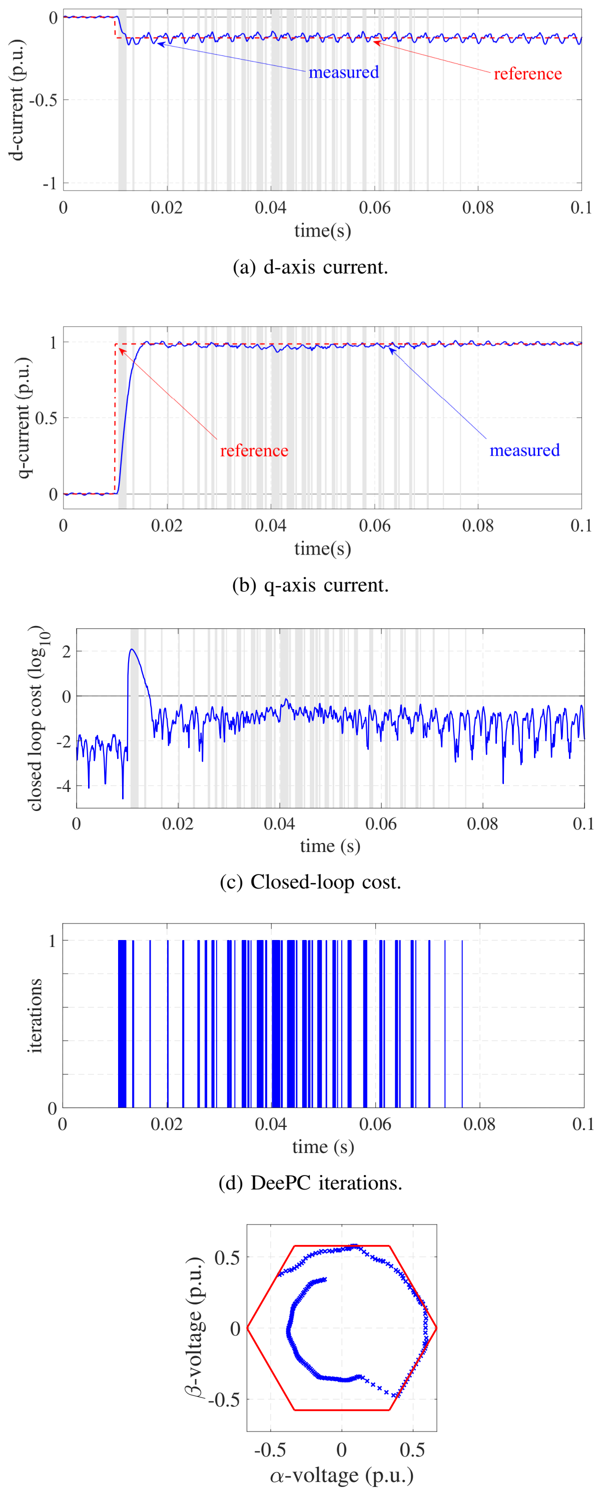
Fig. 11 shows the average turnaround time of the algorithm as a function of the number of iterations of the solver. Currents measurements, position sensing and the Park transformation require about $8\,\mu$s. Thus, the net turnaround time of the constrained DeePC is retrieved by removing $8\,\mu$s to the results of Fig. 11. In any case, the overall computation time is well below the sampling time of $100\,\mu$s. This result is particularly relevant when compared to the data-driven controller described in [7], which was not implementable at such rates.

When the unconstrained DeePC solution is feasible, no further iteration is needed. The computation time grows with the iterations, since the fundamental iterations is repeated. Every iteration involves the computations described in Sec. III-B. The cases when the solutions are found on the vertices, i.e., *iteration* 1.5 and 2.5, are merged to the *iteration* 1 and 2, respectively. In fact, no additional computations are required for these cases [14], meaning that it is not needed to solve problem (14) another time.

E. Controller dynamic performance

This section aims to prove that the proposed data-driven methodology is also capable of achieving very favorable dynamic performances. It is remarked that the work does not focus into optimizing the dynamic of the PMSM drive. Instead, it aims at demonstrating that the problem reduction method allows for enabling the real-time implementation of data-driven controllers, such as the DeePC.

The IPM machine is controlled in current mode, and it is dragged at its nominal speed by means of the load machine. Then, a step-wise change of the current references is commanded, from zero to the nominal maximum-torque-per-ampere value, i.e. $\boldsymbol{i}^*_\mathrm{dq} = (I_\mathrm{N,d},\, I_\mathrm{N,q})^\mathrm{T}$ (see Table I). The d and q current responses are reported in Fig. 12(a) and Fig. 12(b), respectively. As shown in figures, the desired set-point is achieved with just a reduced undershoot on the d-axis. The q-current reference reaches the steady-state in the last



(a) d-axis current.

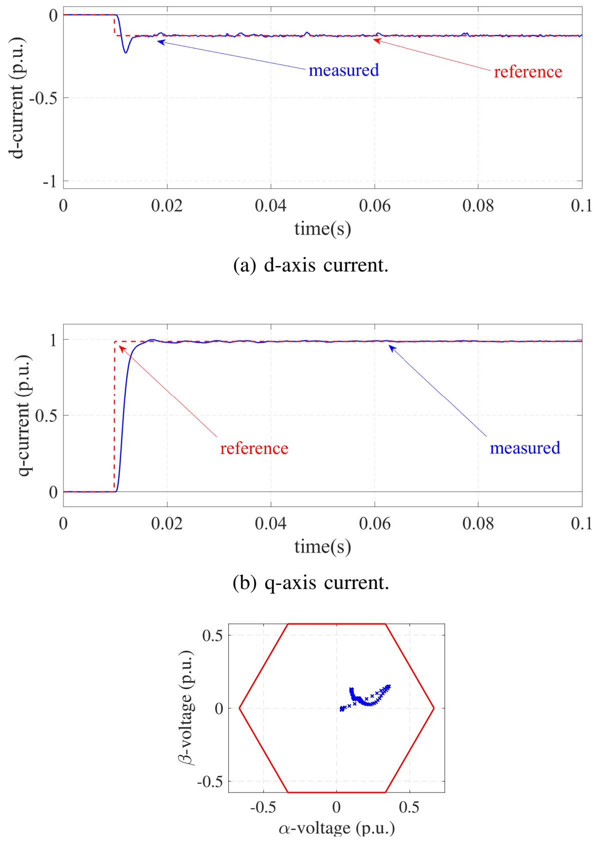
(b) q-axis current.

(c) Closed-loop cost.

(d) DeePC iterations.

(e) Voltages in stator $\alpha\beta$ reference frame.

Fig. 12: Step response analysis at nominal speed.



(a) d-axis current.

(b) q-axis current.

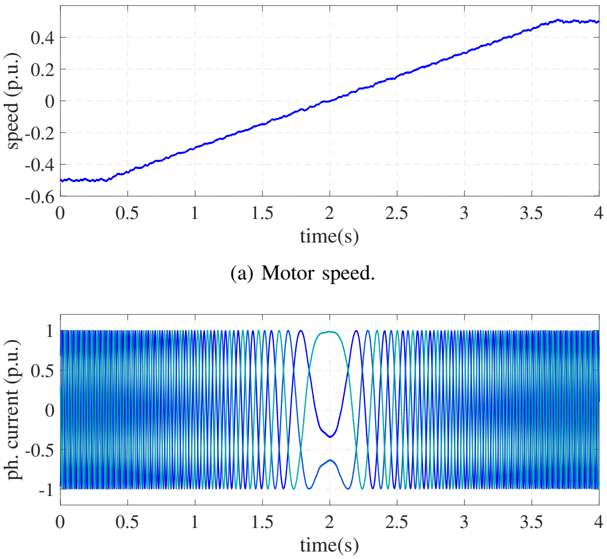
(c) Voltages in stator $\alpha\beta$ reference frame.

Fig. 13: Step response analysis at 10% of the nominal speed.

part of the test because of a instantaneous reduction of the available inverter DC bus voltage. This fact is proven also by the computation of the closed loop cost, reported in Fig. 12(c). This cost is obtained by evaluating $\|\boldsymbol{i}^*_{\mathrm{dq}} - \boldsymbol{i}_{\mathrm{dq}}(k)\|_{\mathbf{Q}} + \|\boldsymbol{u}_{\mathrm{dq}}(k-1)\|_{\mathbf{R}}$ (3) every control period. In the beginning of the cost is slightly higher because of the step-wise change in the current references. Fig. 12(d) further confirms that the solver in the beginning of the test is finding either feasible unconstrained solution or solution that violates one of the hexagon constraints. Thus, the motor is operating near the bounds of the feasible voltage region. Fig. 12(e) reports the inverter voltages in the stator reference frame $\alpha\beta$. In particular, the trajectory walks along the voltage hexagon edges, until the current reaches its reference.

The step response was repeated also at 10% of the nominal speed, and the results are summarized in Fig. 13. Current responses are quite similar to the nominal speed case in terms of rising time. The steady-state solutions of the DeePC problem do never touch the hexagonal set, as shown in Fig.13(c). This is due to the fact that a low voltage vector is sufficient at such low speed to track the desired current reference. If faster transients are required, designers can adjust the control response by reducing $\mathbf{R}$ in (3a). Higher voltage demands are expected is such scenario.

For the sake of completeness, the proposed controller performance is investigated during speed transients, too. In particular, the load motor performs a speed inversion from $+50\%$ to -50% of the rated value (Fig. 14(a)), while nominal



(a) Motor speed.

(b) Phase currents.

Fig. 14: DeePC performance during a speed transient.

currents are commanded to the motor under test. Results in terms of phase current wave-forms are shown in Fig. 14(b). The DeePC demonstrate a robust behavior during the transient regardless the working speed. This behavior is due to the integral action included in the controller (see Section III C of [7]). Future research activities may investigate the possibility to provide an integral action by recursively updating the data set carried by the DeePC.

V. Conclusion

A methodology to reduce the computational burden of constrained data-enabled predictive control of synchronous motor drives is discussed in this work. The proposed solution couples a solver for the specific embedded application and a customized problem reduction technique, based on the proper orthogonal decomposition. Thanks to the proposed method, the constrained data-enabled predictive control has been real-time implemented in a R&D hardware for the first time. The effectiveness of the proposed reduction method is proven by means of simulations. In addition, the problem size reduction is significant and the problem size is no more influenced by the amount of available data. The accuracy of the reduced problem is validated, taking into consideration all the solver iterations. Results are provided to illustrate the benefits of solving the constrained data-enabled control problem, instead of using computationally efficient approximations of the problem. Future works may investigate the possibility to generalize the proposed reduction method for other applications. Moreover, the choice of the parametrization of the control problem could be deeply explored and motivated, e.g. the operating speed may be included as a parameter in the problem reduction procedure. In addition, future activities will address the possibility of building adaptive data-driven controllers. In the area of electric drives, nonlinearties are often found, due to the iron saturation in the electric machines or the presence of dead-times in the power converters. Thus, a data-driven controller will be expected to self-adapt its behavior according to the working condition.

References

[1] M. Schenke, W. Kirchgässner, and O. Wallscheid, “Controller design for electrical drives by deep reinforcement learning: A proof of concept,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4650–4658, 2020.

[2] C. De Persis and P. Tesi, “Formulas for data-driven control: Stabilization, optimality, and robustness,” *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 909–924, 2020.

[3] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, “Decentralized data-enabled predictive control for power system oscillation damping,” *IEEE Trans. Control Syst. Technol.*, pp. 1–13, 2021.

[4] I. Markovsky and F. Dörfler, “Behavioral systems theory in data-driven analysis, signal processing, and control,” *Annual Reviews in Control*, 2021.

[5] J. Rodriguez *et al.*, “Latest advances of model predictive control in electrical drives—part i: Basic concepts and advanced strategies,” *IEEE Trans. Pow. Electron.*, vol. 37, no. 4, pp. 3927–3942, 2022.

[6] A. A. Ahmed, B. K. Koh, and Y. I. Lee, “A comparison of finite control set and continuous control set model predictive control schemes for speed control of induction motors,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1334–1346, 2018.

[7] P. G. Carlet, A. Favato, S. Bolognani, and F. Dorfler, “Data-driven continuous-set predictive current control for synchronous motor drives,” *IEEE Trans. Pow. Electron.*, pp. 1–1, 2022.

[8] P. Karamanakos and T. Geyer, “Guidelines for the design of finite control set model predictive controllers,” *IEEE Trans. Pow. Electron.*, vol. 35, no. 7, pp. 7434–7450, 2020.

[9] H. Xie, F. Wang, Q. Xun, Y. He, J. Rodriguez, and R. Kennel, “A low-complexity gradient descent solution with backtracking iteration approach for finite control set predictive current control,” *IEEE Trans. Ind. Electron.*, pp. 1–1, 2021.

[10] A. A. Ahmed, B. K. Koh, and Y. I. Lee, “A comparison of finite control set and continuous control set model predictive control schemes for speed control of induction motors,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1334–1346, 2018.

[11] Y. Yang, H. Wen, R. Chen, M. Fan, X. Zhang, M. Norambuena, and J. Rodriguez, “An efficient model predictive control using virtual voltage vectors for three-phase three-level converters with constant switching frequency,” *IEEE Trans. Ind. Electron.*, pp. 1–1, 2021.

[12] G. Cimini, D. Bernardini, S. Levijoki, and A. Bemporad, “Embedded model predictive control with certified real-time optimization for synchronous motors,” *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 893–900, 2021.

[13] S. Hanke, O. Wallscheid, and J. Böcker, “Continuous-control-set model predictive control with integrated modulator in permanent magnet synchronous motor applications,” in *2019 IEEE International Electric Machines Drives Conference (IEMDC)*, 2019, pp. 2210–2216.

[14] A. Favato, P. G. Carlet, F. Toso, R. Torchio, L. Ortombina, M. Bruschetta, R. Carli, P. Alotto, S. Bolognani, and J. Rodriguez, “Fast solver for implicit continuous set model predictive control of electric drives,” *IEEE Access*, pp. 1–1, 2022.

[15] A. Chatterjee, “An introduction to the proper orthogonal decomposition,” *Current Science*, vol. 78, no. 7, pp. 808–817, 2000. [Online]. Available: http://www.jstor.org/stable/24103957

[16] K. Kunisch and S. Volkwein, “Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics,” *SIAM Journal on Numerical Analysis*, vol. 40, no. 2, pp. 492–515, 2002. [Online]. Available: https://doi.org/10.1137/S0036142900382612

[17] C. Jia, X. Wang, Y. Liang, and K. Zhou, “Robust current controller for IPMSM drives based on explicit model predictive control with online disturbance observer,” *IEEE Access*, vol. 7, pp. 45 898–45 910, 2019.

[18] J. Coulson, J. Lygeros, and F. Dorfler, “Distributionally robust chance constrained data-enabled predictive control,” *IEEE Trans. Autom. Control*, pp. 1–1, 2021.

[19] F. Toso, P. G. Carlet, A. Favato, and S. Bolognani, “On-line continuous control set MPC for PMSM drives current loops at high sampling rate using qpOASES,” in *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2019, pp. 6615–6620.

[20] D. K. M. Kufoalor, B. J. T. Binder, H. J. Ferreau, L. Imsland, T. A. Johansen, and M. Diehl, “Automatic deployment of industrial embedded model predictive control using qpOASES,” in *2015 European Control Conference (ECC)*, 2015, pp. 2601–2608.

[21] A. Favato, P. G. Carlet, F. Toso, R. Torchio, L. Ortombina, M. Bruschetta, R. Carli, S. Bolognani, and J. Rodriguez, “Efficient QP solver for electric motors,” *TechRxiv*, 2021.

[22] G. Rozza, D. B. P. Huynh, and A. T. Patera, “Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations,” *Archives of Computational Methods in Engineering*, vol. 15, no. 3, p. 229, May 2008. [Online]. Available: https://doi.org/10.1007/s11831-008-9019-9

[23] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu, “Proper orthogonal decomposition and its applications—part i: Theory,” *Journal of Sound and Vibration*, vol. 252, no. 3, pp. 527–544, 2002.

[24] H. Ferreau, H. Bock, and M. Diehl, “An online active set strategy to overcome the limitations of explicit MPC,” *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.