Direct data-driven design of sparse controllers

Simone Formentin and Alireza Karimi

Abstract—This paper deals with direct data-driven design of model-reference controllers whose number of parameters is constrained. Input-output (I/O) sparse controllers are introduced and proposed as an alternative to low-order controller tuning. The optimal I/O sparse controller is shown to be never worse than the optimal low-order controller with the same number of parameters and a suited design procedure based on convex optimization is derived. The theoretical concepts are illustrated by means of a benchmark simulation example.

I. INTRODUCTION

In the industrial practice, the main problems related to high-complexity controller implementation are essentially two:

1) when the number of parameters is large, many arithmetic operations, i.e., multiplications and additions, are required, thus slowing down the processing;

2) high-complexity controllers are fragile, i.e., highly sensitive to roundoff errors (see [5]).

Based on the previous observations, many important contributions in control theory have appeared, reaching from classical works like, e.g., Ziegler and Nichols methods, to more recent papers where low-complexity controller design is dealt with following three main approaches. Firstly, low-complexity controllers can be designed starting from a complex model of the system, by performing model order-reduction (see [11] for a survey on the topic) and then designing a controller based on the simplified model. Alternatively, high-complexity controllers derived from high-complexity models can be approximated with low-complexity ones by means of controller order-reduction (see now [1] for an overview). Finally, fixed-order controllers can be tuned from high-complexity models without explicit order-reduction, see e.g., [9]. Inside the latter category, a special place is occupied by the so-called “direct data-driven methods”, where a model of the system is not needed at all and a fixed-order controller is directly derived from input-output (I/O) data (see [2]).

This work will focus on noniterative direct data-driven techniques and will show that, in this framework, low-order controller design is neither the only nor the best way to find a controller with fixed number of parameters. The main idea can be explained by means of a simple example. Consider that a $n^{th}$-order FIR controller is available for a given application, but only $n_p < n$ parameters are allowed for numerical reasons. The standard approaches presented above would either reduce the order of the FIR down to $n_p$ to comply with the problem specifications, or directly compute a $n^{th}$-order FIR controller. However, one should notice that a suited controller with the assigned number of parameters could be also found among the FIR controllers of order $n$, where only $n_p$ elements are allowed to be different than zero (i.e., where the parameter vector is “sparse”). This set of controllers include the $n^{th}$-order FIR, then the final result, i.e., the controller in this set achieving the best performance, cannot be worse than that. The only drawback in using such a “sparse” solution is that more RAM memory is needed for on-line use, because the number of input samples required by the controller is greater than $n_p$. Nevertheless, this is definitely not a real problem dealing with modern technology and its (low) cost.

In this paper, the concept of input-output (I/O) sparse controllers will be formally defined and proposed as a better way to deal with “controller simplification”. As an example, the Correlation-based Tuning approach (see [14]) will be treated without loss of generality. The same discussion could be straightforwardly extended to other techniques based on linear controller parameterization, like e.g. Virtual Reference Feedback Tuning (VRFT [3]), [7] or [9].

It should be mentioned that controllers with sparse state-space matrices are used in distributed control to reduce the number of needed channels and the computational burden, see e.g. [6] and references therein. Few studies have been presented also in “non-fragile” controller design, to enhance robustness of the closed-loop properties against numerical implementation problems using sparse state-space matrices for the controller (see e.g. [15] and references therein).

However, as far as the authors are aware, the concept of sparse controllers in I/O description and the idea of using them as an alternative to low-order controllers have never been presented in the scientific literature. Moreover, it is the first time that sparsity is used in the data-driven controller tuning framework.

The outline of the paper is as follows. In Section II, the study of sparse I/O controllers is motivated as an alternative to low-order controllers by means of a benchmark example. In Section III, an extension for the direct data-driven technique in [14] to sparse controller tuning is then proposed, using convex optimization. This solution is suboptimal with respect to the best $H_2$ sparse controller with assigned number of parameters, but more practicable and still better than the one given by the standard approach. Some final remarks end the paper in Section V.

II. MOTIVATION

Consider the flexible transmission system introduced as a benchmark for digital control design in [10]. The plant is...
described by the discrete-time model
\[ G(q^{-1}) = \frac{0.28261q^{-3} + 0.50666q^{-4}}{a(q^{-1})} \quad (1) \]
where \( a(q^{-1}) = 1 - 1.41833q^{-1} + 1.05839q^{-2} - 1.31608q^{-3} + 0.88642q^{-4} \) and \( q^{-1} \) is the backward shift operator. For simplicity, consider that no noise affects the output measurements. The control objective is given in terms of model-reference specifications, where the reference model reads (see [3])
\[ M(q^{-1}) = \frac{(1 - \alpha)^2 q^{-3}}{(1 - \alpha q^{-1})^2}; \quad \alpha = e^{-0.5}. \quad (2) \]
Let the quality of the closed-loop model matching when a fixed-order controller \( K(q^{-1}, \rho) \), parameterized through \( \rho \), is inserted in the loop, be evaluated by means of the \( \mathcal{H}_2 \)-cost
\[ J_{mr}(\rho) = \left\| \begin{bmatrix} G(K(\rho)) \\ 1 + GK(\rho) \end{bmatrix} - M \right\|_2^2. \quad (3) \]
Specifically, suppose that the \( n^{th} \)-order controller is linearly parameterized, as
\[ K(q^{-1}, \rho) = \beta^T(q^{-1}) \rho \quad (4) \]
where the basis functions are \( \beta_i(q^{-1}) = q^{-i}/(1 - q^{-1}) \), \( i = 0, \ldots, n - 1 \) and the number of parameters potentially different than zero, namely \( n_\rho \), coincide with the order \( n \). Notice that there is no controller with given structure (4) that allows (2) to be achieved. Suppose now that some computational constraints are given, such that only up to 3 parameters can be used in the controller. Standing on the assumption that a set of persistently exciting data is provided, the 2\( n^{th} \)-order controller
\[ K_{CBT}(q^{-1}) = \frac{0.2208 - 0.3326q^{-1} + 0.1992q^{-2}}{1 - q^{-1}} \quad (5) \]
can be easily computed using the direct data-driven controller tuning methods in [14]. In Fig. 1 and 2, the poor results given by use of \( K_{CBT} \) are illustrated (red dashed line).
Let us now consider the controller
\[ K^*(q^{-1}) = \frac{0.1571 - 0.1044q^{-1} + 0.0811q^{-4}}{1 - q^{-1}}. \quad (6) \]
In Fig. 1 and 2, it is shown that this controller clearly outperforms \( K_{CBT} \), even if it is still characterized by \( n_\rho = 3 \) parameters, because now the order is higher, i.e. 4.
From now on, all controllers like \( K^* \) are said “I/O sparse”, according to the following definition.

\textbf{Definition 1}: Let \( \mathcal{K}^n \) be the set of all controllers of order \( n \), parameterized as in (4). A controller \( K \in \mathcal{K}^n \) is said “I/O sparse” if the number of non-zero elements of \( \rho \), namely \( n_\rho \), is \( n_\rho \ll n \).

Notice that, since the “I/O sparse” parameterization include the low-order one, the optimal sparse solution \textit{can never be worse} than the best low-order controller with the same \( n_\rho \).
The previous example can be discussed as follows:

- the key issue is the number of parameters and not the order of the controller. Therefore, \textit{better performance can be achieved, by keeping the constraint on \( n_\rho \) and relaxing the standard consideration \( n_\rho = n \).}
- The number of arithmetic operations required by \( K^* \) and \( K_{CBT} \) for on-line data processing is the same.
- With the parameterization (4), the problem of finding the best 3-parameter controller is no longer a problem of order reduction of the controller achieving \( M \), whose solution is well-known in the literature, but it becomes more complicated.

The latter (innovative) design problem will be faced in the following section.
III. EXPLOITING SPARSITY IN DIRECT DATA-DRIVEN CONTROL DESIGN

A. Backgrounds

Correlation-based Tuning (CbT, see [14]) is a recent non-iterative direct data-driven technique devoted to the design of model-reference controllers without need of a model of the plant. The complete procedure relies only on one convex optimization step, thus guaranteeing to provide the global minimum of the approximate model-reference cost, defined as follows. Let us assume that the sensitivity function of the closed-loop system with the estimated controller \( K(\hat{\rho}_N) \) is a good approximation of the ideal one, that is

\[
1 + GK(\hat{\rho}_N) \approx 1 - M. 
\]

It follows that (3) shares the same minimum of a new cost function, convex in \( \rho \), defined as

\[
J(\rho) = \|(K(\rho)(1 - M)G - M)(1 - M)\|^2. 
\]

Such a minimum is also the minimum of the \( L_2 \)-norm of the matching error signal \( \varepsilon_c \) in Fig. 3, when the signals are noiseless and \( u(t) \) is a white noise of unit variance used as reference signal, namely \( r(t) = u(t) \). The most important observation at the basis of the CbT rationale is that, in the noiseless setting, the model matching error \( \varepsilon_c(t, \rho) \) can be directly computed from I/O data as follows:

\[
\varepsilon_c(t, \rho) = Mu(t) - (1 - M)K(\rho)Gr(t) \\
\varepsilon_c(t, \rho) = Mu(t) - (1 - M)K(\rho)y(t)
\]

and the minimizer of the \( L_2 \)-norm of \( \varepsilon_c(t, \rho) \) is exactly the minimizer of (8).

When data are collected in a noisy environment, the method resorts to the correlation approach to identify the controller. Specifically, an extended instrumental variable \( \xi(t) \) correlated with \( u(t) \) and uncorrelated with \( v(t) \) is introduced to decorrelate the error signal \( \varepsilon_c(t, \rho) \) and \( u(t) \). \( \xi(t) \) is defined as

\[
\xi(t) = [u(t + l), \ldots, u(t), \ldots, u(t - l)]^T.
\]

where \( l \) is a sufficiently large integer. The correlation function is defined as

\[
f_{N,l}(\rho) = \frac{1}{N} \sum_{t=1}^{N} \xi(t)\varepsilon_c(t, \rho) 
\]

and the correlation criterion as

\[
J_{N,l}(\rho) = f_{N,l}^T(\rho)f_{N,l}(\rho). 
\]

It follows that the optimal \( H_2 \) model-reference controller is the solution of the convex optimization problem

\[
\hat{\rho} = \arg \min_{\rho} J_{N,l}(\rho). 
\]

In [14], it has been proven that the method is consistent for any input sequence, if data in \( \xi(t) \) are prefiltered by \( L(q^{-1}) \), defined as

\[
L(q^{-1}) = \frac{1 - M(e^{-j\omega})}{\Phi_u(\omega)},
\]

where \( \Phi_u(\omega) \) denotes the spectral density of \( u(t) \). Notice that such a prefilter may be non-causal but it can be implemented off-line.

The fact that \( K(\hat{\rho}_N) \) is a minimizer of the control cost does not give any guarantees about the internal stability of the system. Instability can occur if \( M \) is unachievable with the given controller parameterization or the variance of the estimate due to finite length \( N \) of the dataset is large. In order to guarantee that \( K(\hat{\rho}_N) \) is stabilizing, in [14] it has been shown that the optimization procedure has to be constrained by the frequency-wise set of equations

\[
\left| \sum_{\tau=-l_2}^{l_2} \hat{R}_{\tau e}(\tau, \rho)e^{-j\tau\omega_k} \right| \leq \left| \sum_{\tau=-l_2}^{l_2} \hat{R}_e(\tau)e^{-j\tau\omega_k} \right|,
\]

where

\[
\hat{R}_{\tau e}(\tau, \rho) = \frac{1}{N} \sum_{t=1}^{N} r(t - \tau)\varepsilon_c(t, \rho), \quad \tau = -l_2, \ldots, l_2,
\]

\[
\hat{R}_e(\tau) = \frac{1}{N} \sum_{t=1}^{N} r(t - \tau)r(t), \quad \tau = -l_2, \ldots, l_2
\]

and \( l_2 \) defines the length of a rectangular window, that also determines the resolution of the discrete frequency axis \( \omega_k \). Notice that (14) is a set of convex constraints, as (15) is linear in \( \rho \), and then the overall optimization problem remains convex. It should be here recalled that the constraint introduce some conservatism into the solution, as it is based on a Small-Gain condition that is only sufficient for guaranteeing the internal stability property (see [14]).

B. Sparse controller tuning using \( L_1 \)-regularization

Suppose now that \( n_\rho \) is fixed. According to the rationale of this work, there might be a combination of \( n_\rho \) basis functions \( \beta_j \) (out of a set of \( n > n_\rho \)) such that the model-matching performance is better than that given by the \( n_\rho \)th order controller, i.e. such that the value of (3) is lower. Notice that, theoretically speaking, \( n \) has no limits, but in practice an upper bound is given by the maximum order allowed, namely \( n_{max} \), depending on the RAM memory availability in controller implementation.

A straightforward way to put to zero some elements of the parameter vector \( \rho \) is to weight the number of non-zero elements inside the cost function. Formally, replace (11) with

\[
\tilde{J}_{N,l}(\rho) = f_{N,l}^T(\rho)f_{N,l}(\rho) + \lambda \|\rho\|_0, \quad \lambda \in \mathbb{R}^+,
\]
where \( \|\cdot\|_0 \) denotes the \( \mathcal{L}^0 \)-norm, i.e. the number of non-zero elements of a vector \( 1 \), and the dimension of \( \rho \) is equal to the maximum order allowed \( n_{\text{max}} \). Depending on the value of \( \lambda \), the number of zero elements will vary and then \( \|\rho\|_0 = n_{\rho} \) can be fixed by suitably selecting \( \lambda \). This is by no means a new concept when talking about sparsity in optimization or system identification. Unfortunately, it is also well known that \( \mathcal{L}^0 \)-norm makes the optimization problem non-convex and can be proven NP-hard (see [8]). However, since the 1970s, \( \mathcal{L}^1 \)-norm is used instead of \( \mathcal{L}^0 \)-norm to obtain a convex relaxation of (17) yielding good results (see [13] for a historical survey). In this work, an iterative strategy inspired from [4] will be employed. For a given \( \lambda \), the procedure is summarized in the following algorithm.

### CbT with weighted \( \mathcal{L}^1 \) regularization for a given \( \lambda \).

1. set \( n_{\rho}, \varepsilon > 0 \) and the maximum number of iterations allowed \( j_{\text{max}} \);
2. set the iteration counter \( j = 0 \) and the weighting diagonal matrix \( W^{(0)} \) equal to the identity matrix;
3. solve the mixed \( \mathcal{L}^1 \)-\( \mathcal{L}^2 \) weighted minimization problem (subject to stability constraint)
   \[
   \hat{\rho}^{(j)} = \arg\min_\rho \left\{ f_{\mathcal{L}^1}(\rho)f_{\mathcal{L}^2}(\rho) + \lambda \left( \| W^{(j)} \rho \|_1 \right) \right\};
   \text{s.t. (14)} \tag{18}
   \]
4. update the diagonal elements \( w_i \) of the weighting matrix \( W \) as:
   \[
   w_i^{(j+1)} = \frac{1}{\| \rho_i^{(j)} \| + \varepsilon}, \quad i = 1, \ldots, n_{\text{max}}, \tag{19}
   \]
   where \( \rho_i \) is the \( i \)-th element of \( \rho \);
5. terminate the iterations on convergence or when \( j = j_{\text{max}} \). Otherwise, go to step 3.

If the number of non-zero parameters is equal to \( n_{\rho} \), the procedure terminates. Otherwise, another \( \lambda \) in a given interval \([0, \lambda_{\text{max}}]\) must be selected (the way \( \lambda_{\text{max}} \) is chosen will be discussed later in this Section). In this paper, the following procedure for \( \lambda \) will be employed. First of all, a step-size \( \Delta \lambda = \lambda_{\text{max}}/n_\lambda \) is selected, where \( n_\lambda \) can be arbitrarily large. Then, if the number of non-zero parameters is larger than \( n_{\rho} \), the actual \( \lambda \) is simply augmented of \( \Delta \lambda = \lambda_{\text{max}}/n_\lambda \) and the CbT algorithm is restarted with the same \( n_{\rho} \). If the number of non-zero parameters is smaller than \( n_{\rho} \), a new interval is considered for the choice of \( \lambda \), where the largest value is the actual one and the smallest is the one at the previous iteration. A new step-size is created and \( \lambda \) is gradually augmented until \( n_{\rho} \) is reached. Notice that the previous approach guarantees to have the smallest \( \lambda \) for a given \( n_{\rho} \), once the resolution \( n_\lambda \) is provided.

The reason why one should use a weighted version of the \( \mathcal{L}^1 \)-norm instead of simple \( \mathcal{L}^1 \)-regularization is quite simple. Assume that the algorithm converges to \( \rho^* \) in \( j \) steps and that \( \varepsilon \) is at least one order of magnitude smaller than the smallest element of \( \rho^* \). Then, at limit:

\[
\lim_{j \to j} \left\| W^{(j)} \rho^{(j)} \right\|_1 = \lim_{j \to j} \sum_{i=1}^{n} w_i^{(j)} \left| \rho_i^{(j)} \right| = \lim_{j \to j} \sum_{i=1}^{n} \left| \rho_i^{(j-1)} \right|^{(j)} + \varepsilon,
\]

that is, using the above hypotheses,

\[
\lim_{j \to j} \left\| W^{(j)} \rho^{(j)} \right\|_1 \approx \sum_{i=1}^{n} \left| \text{sign}(\rho_i^{(j)}) \right| = \| \rho^* \|_0. \tag{21}
\]

In other words, the weighted \( \mathcal{L}^1 \)-norm is a better approximation of the \( \mathcal{L}^0 \)-norm than standard \( \mathcal{L}^1 \)-norm (for further details on this topic, see [4]).

Notice that this algorithm requires the tuning of three additional parameters with respect to the standard strategy, i.e. \( \varepsilon, j_{\text{max}} \) and \( \lambda_{\text{max}} \), that have not been discussed so far. However, it is known from [4] that the method is robust to the choice of \( \varepsilon \); further, \( j_{\text{max}} \) is fully determined by computational constraints. It follows that a deeper insight is required only for the selection of \( \lambda_{\text{max}} \). The following result provides some guidelines about it.

**Proposition 1:** At each iteration \( j \), if \( \lambda \geq \lambda_{\text{max}}^{(j)} \), where \( \lambda_{\text{max}}^{(j)} \) is defined as in (22), then the minimizer of (18) is identically 0, where 0 represents the vector of n zeros.

**Proof:** The proof follows the same line of the proof of Proposition 4.1. in Chapter 4 of [12]. A necessary and sufficient condition for 0 to be solution of (18) is that it belongs to the interval given by the set of sub-gradients of (18) at \( \rho = 0 \). For each element of the parameter vector \( \rho_i \), \( i = 1, \ldots, n_{\text{max}} \), the latter is

\[
\mathcal{I}_{\rho_i} = \left[ \frac{\partial J_{\mathcal{L}^1}(\rho)}{\partial \rho_i} \right]_{\rho=0} - \lambda w_i^{(j)} \left[ \frac{\partial J_{\mathcal{L}^1}(\rho)}{\partial \rho} \right]_{\rho=0} + \lambda w_i^{(j)}, \tag{23}
\]

then to make \( \rho = 0 \) an optima, \( \lambda \) must be such that

\[
\lambda \geq \frac{1}{w_i^{(j)}} \left| \frac{\partial J_{\mathcal{L}^1}(\rho)}{\partial \rho_i} \right|_{\rho=0}, \quad i = 1, \ldots, n_{\text{max}}. \tag{24}
\]

Since

\[
\frac{\partial J_{\mathcal{L}^1}(\rho)}{\partial \rho_i} \bigg|_{\rho=0} = 2 \left[ \frac{1}{N} \sum_{i=1}^{N} \zeta(t) \beta_i(1-M)y(t) \right]^T \times \frac{1}{N} \sum_{i=1}^{N} \zeta(t)Mu(t), \quad i = 1, \ldots, n_{\text{max}},
\]

Equation (22) follows.

The previous result provides a hint for the \( j \)-th iteration, but it does not give guarantees for the whole procedure (for

---

\(^1\)Since this definition does not satisfy the homogeneity property of a norm, it is called a quasi-norm as well.
\[
\lambda^{(j)}_{\text{max}} = \left\| \frac{2}{\inf_{i} u_i^{(j)}} \left[ \frac{1}{N} \sum_{t=1}^{N} \zeta(t) \beta^{T}(1 - M)y(t) \right]^{T} \frac{1}{N} \sum_{t=1}^{N} \zeta(t)M u(t) \right\|_{\infty}
\]  \quad (22)

which \( \lambda \) is unique). However, \( \lambda^{(j)}_{\text{max}} \) can still be used as a check variable, that, when overcome, can stop the procedure to select a lower \( \lambda \) at step 1 of the algorithm.

C. Optimal I/O sparse controller

Notice that this procedure is not optimal, because the best sparse approximation of the controller achieving \( M \) is not a minimizer of (11), but it is given by (18). The optimal solution is instead discussed below.

Since \( n_{\text{max}} \) is known (determined by memory bounds) and \( n_p \) is given, the number of possible combinations \( m \) of \( \beta_i \) can be computed exactly as

\[
m = \frac{n_{\text{max}}!}{n_p!(n_{\text{max}} - n_p)!} \quad (25)
\]

Theoretically speaking, the computation of the optimal controller is then feasible and can be found by evaluating the closed-loop model-matching performance of all the \( m \) controllers and choosing the one that gives the lowest value of \( J_{m_{\text{opt}}} \). However, this approach may have two practical problems:

1) \( m \) increases very fast with \( n_{\text{max}} \), that means that the computational time required by the method may no longer be negligible. As an example, see Fig. 4, where \( n_p = 3 \) like in the example of Section II and \( m \) is plotted as a function of \( n_{\text{max}} \).

![Fig. 4. \( m \) as a function of \( n_{\text{max}} \), if \( n_p = 3 \).](image)

2) To evaluate the effect of each controller on (3), the only way is to assess its performance on (11), since \( G \) is not available. However, (11) is (asymptotically) the data-driven counterpart of (8), then it shares the same minimum of (3) only in case (7) holds. This means that, in all the other situations, not only the performance may decrease but also the evaluation task may become hard to do, unless one perform one experiment on the plant for each controller!

The suboptimal (but practicable) solution proposed in Subsection III-B is then a nice an alternative to this optimal sparse \( \mathcal{H}_2 \) design strategy in many practical situations. Moreover, it has several interesting properties:

- it requires just one experiment;
- for each iteration, only a convex optimization problem has to be solved, even if the overall procedure aims at approximating the solution of a non-convex problem;
- if \( \lambda \) is small, solutions of (18) and (11) are very close to each other (and the method is only slightly suboptimal).

IV. SIMULATION EXAMPLE - REVISITED

Consider again the flexible transmission system of Section II. With the same dataset used for tuning the 2\textsuperscript{nd}-order controller, it is possible to design a sparse controller with 3 parameters according to the theory presented in the previous section. In detail, if it is assumed that \( n_{\text{max}} = 12 \), the number of combinations is \( m = 220 \) and the best selection is given by the “mysterious” controller \( K^* \), used in Section II. Its performance, together with that of the 2\textsuperscript{nd}-order controller, is reported again in Fig. 5 and 6 and in Table I. In the same diagrams and table, the results obtained using weighted \( L_1 \)-regularization are also shown. Specifically, given \( j_{\text{max}} = 10 \), \( \varepsilon = n_{\lambda} = 0.001 \) and the minimum \( \lambda \) yielding 3 parameters in 10 iterations, \( i.e. \lambda = 0.007 \), the controller

\[
K_{L_1}(q^{-1}) = \frac{0.0779 - 0.0114q^{-1} + 0.0561q^{-5}}{1 - q^{-1}}, \quad (26)
\]

is obtained. From Table I, it can also be checked that, even if the maximum \( j \) is 10, the iterative procedure converges to 3 parameters in only 5 iterations. In all the simulations, according to Proposition 1, a check that \( \lambda \) was not going beyond the maximum value allowed was done. Being \( \lambda \) small, this never happened and the procedure was not forced to restart.

Notice that the difference between the frequency response of the complementary sensitivity function using \( K^* \) and the one using \( K_{L_1} \) is large only at high frequencies, while it remains small before the first resonance peak. It should be here stressed that, no matter how the solution is computed, both the considered sparse controllers (\( \mathcal{H}_2 \) optimal and suboptimal with weighted \( L_1 \)-regularization) are better than the 2\textsuperscript{nd}-order controller computed using the standard method. In all the cases, the stability constraint (14) was not active, then the final performance was not decreased due to its conservatism.

Remark. For sake of completeness, a \( L_1 \) controller could be computed by simply minimizing (18) in one step (where \( W \) is the identity matrix). If the same procedure for the selection of \( \lambda \) is used, the controller

\[
K_{\text{std}}(q^{-1}) = \frac{0.0731 + 0.0470q^{-5} + 0.0075q^{-6}}{1 - q^{-1}}, \quad (27)
\]
for $\lambda = 0.1$ is obtained. The assessment of closed-loop performance provided by (27) are in Table II, where it is shown that model-matching error is larger than the one yielded by the weighted version of the regularized estimate. Notice that this result was expected from [4], since the weighted $L_1$ regularization is a more accurate approximation of the $L_0$ problem.

V. CONCLUSIONS

In this paper, I/O sparse controllers have been proposed as an alternative to low-order controllers in the data-driven control framework. It has been shown that the best $H_2$ sparse approximation of the optimal controller can be found from data by means of combinatorial (and convex) optimization, however, this method is not suited if the set of candidate controllers is large or the number of desired parameters is much lower than the one characterizing the optimal controller. For these situations, a different method using weighted $L_1$-regularization has been introduced and some hints on how to select the main tuning knobs have been given. The method proved effective on the benchmark example introduced in [10]. Future work will focus on extension of I/O sparse controllers to other control design methods.

REFERENCES