Linear Algebra and its Applications 644 (2022) 149–171

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

A linear algorithm for the minimal realization problem in physical coordinates with a non-invertible output matrix

Chiara Faccio [a], Fabio Marcuzzi [b,*]

[a] *Scuola Normale Superiore, Piazza dei Cavalieri 7, 56126 Pisa, Italy*
[b] *Department of Mathematics "Tullio Levi Civita", University of Padova, Via Trieste 63, 35131 Padova, Italy*

ARTICLE INFO

Article history:
Received 5 August 2021
Accepted 28 February 2022
Available online 4 March 2022
Submitted by V. Mehrmann

MSC:
93B15
15A29
37M99
65F30
93B20
93B30

Keywords:
Subspace identification
Physical parameters estimation
LTI systems
Minimal realization

ABSTRACT

In this paper we present a linear algorithm that estimates some physical parameters of a continuous-time system, described by an analytical mathematical model, when not all the state variables can be measured. The algorithm starts from the well-known subspace methods and applies some linear transformations to recover, at least partially, the estimated model in physical coordinates. Some analytical investigations and numerical experiments are shown for this method, which has general application within linear time-invariant (LTI) dynamical systems.

* Corresponding author.
E-mail address: marcuzzi@math.unipd.it (F. Marcuzzi).

https://doi.org/10.1016/j.laa.2022.02.038
0024-3795/

1. Introduction

Let us consider a general, finite-dimensional, deterministic, Discrete-time, Linear Time-Invariant (DLTI) dynamical system, in the so-called *state-space* form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{1}$$

where $x(k) \in \mathcal{R}^{n_x}$ is the *state vector*, $u(k) \in \mathcal{R}^m$ the *input vector*, $y(k) \in \mathcal{R}^p$ the *output vector* and $A \in \mathcal{R}^{n_x \times n_x}$, $B \in \mathcal{R}^{n_x \times m}$, $C \in \mathcal{R}^{p \times n_x}$ and $D \in \mathcal{R}^{p \times m}$ are the model matrices.

Then, the *minimal (state-space) realization problem* for DLTI systems, can be formulated as follows [1]: "Given some input-output data $u(k), y(k)\ ,\ k = 0, \ldots, N$ about the system (1), find a state-space description of minimal size n_x that is capable of reproducing the given data". The data are typically the impulse response of the system or the step response, or input–output measurements, etc.

The first algorithm for this problem has been developed by Ho and Kalman [2] [3] in 1966, for single-input-single-output (SISO) state-space models. Nowadays, for general multi-input-multi-output (MIMO) state-space models, the solution algorithms for the minimal realization problem have evolved into the so-called "subspace methods" [4] [5] [6]. These algorithms are designed to face also with the presence of noise in the data, which is a core problem in system identification, but the minimal realization problem focuses on algebraic issues, in an ideal, noiseless, settings, like we will use here.

The minimal realization is not unique: given an invertible basis-change matrix T, the system (1) transformed in the new coordinates $\tilde{x} = T^{-1}x$ maintains the same input-output behavior. Since there are infinite invertible matrices T that can be used to change the basis in which to express the state vector, there are also infinite systems that can describe as well the input-output data.

In this paper we deal with the minimal realization problem of systems that are the discretization of a physical-mathematical model; in these systems the state variables have a physical meaning (temperatures, displacements, velocities, etc). Here, then, we try to solve a harder problem: to find a minimal realization whose state vector is expressed in the *physical base*, that is true when each of its state variables has a twin variable in the physical-mathematical model from which the input-output data are supposed to come.

This is an open problem, unless all the state variables have an invertible linear relation with the system's outputs, i.e. the output matrix C is square and invertible. This is seldom the case in practice, and here we consider a more common situation where the model has a rectangular output matrix $C \in R^{p \times n_x}$, $p \ll n_x$, i.e. the number of state variables is bigger than the number of the outputs. The difficulty to get a minimal realization in the physical base is intrinsic to subspace methods. Indeed, they work with linear algebraic operations on the input-output data to produce the minimal realization model matrices $\{A, B, C, D\}$, that are computed with respect to a data-driven base. Hence, our prob-

lem becomes that of finding a basis-change matrix T that transforms the state vector of the minimal realization found by subspace methods in the physical base, among the infinite possible T. In this general formulation, the problem is underdetermined and at our knowledge nobody has found a solution. In this paper we propose a linear algorithm to compute a minimal realization whose state vector is at least partially expressed in the physical base, in a sense that will be precised later in sec. 3.1. The proposed algorithm starts from the minimal realization found by a subspace method and uses linear algebra and also the (nonlinear) discretization map applied to the continuous-time model, to get an estimate of at least some of its (physical) parameters. It turns out to be of general applicability for DLTI systems, like subspace methods are.

Modeling a system with a state-vector expressed in the physical base is of fundamental importance e.g. when the estimation of the physical parameters of the real system is of interest, as will be shown in sec. 3.3. There is a huge literature and interest in physical (grey-box) modeling, in applied sciences, in engineering applications, often involving complex mathematical problems like e.g. computational inverse problems [7] [8]. A well known approach for computing physical parameter estimates is to adopt nonlinear estimation procedures [9]; these, anyway, suffer from convergence problems, depending much from the initialization of the estimates [10] and from the ill-conditioning of the problem to be numerically solved. There is also a wide bibliography on the estimation of physical parameters with subspace methods which aims at estimating all the parameters together: [11] imposes constraints on the impulse response; [12] imposes constraints on the coefficients of the transfer function of the discrete model, [6] imposes constraints on the unknown matrices in the data-equation characterizing subspace identification methods, such as the lower triangular block-Toeplitz of weighting matrices constructed from the Markov parameters of the unknown observer, [13] [14] solve a null-space-based problem. There is also a huge literature on more physical approaches, tailored on a specific application, e.g. in vibration mechanics it well known the so-called “inverse vibration problem”, see e.g. [15], [16]. In the literature there are also methods that exploit the structure of the matrices in the continuum-time model, because of their physical meaning, to compute directly a matrix T which should transform in physical coordinates the estimated model, see [13] or [14] for recent results and further references; to obtain this it is necessary to reformulate the abstract model equations into a null-space-based problem, that brings to a quite involved solution and has in general an high computational cost or it is restricted to a small number of model structures. In this paper, instead, we deal with the problem of estimating only the parameters more tied to the measurable state variables. As we will see, this brings to a considerable simplification of the solution, thus obtaining a linear method, that can be used e.g. also in embedded computing applications, where real-time requirements and tight computing resources demand for efficient numerical methods.

The paper organization is as follows: in sec. 2 we setup the problem and the known methods relevant for this work. In sec. 3 we present a novel strategy to transform a model, obtained from subspace methods, at least partially in the physical base. In sec. 4

we extend this strategy to non-diagonalizable matrices. In sec. 5 we will show the results of a bunch of random tests and then it follows the conclusions.

2. Problem settings

Let us suppose that the real system can be adequately modeled by an analytical mathematical model, expressed as a system of first-order, ordinary differential equations and linear output relations, i.e. an LTI system in state-space form:

$$\begin{array}{rcl} \dot{x}(t) & = & A_c x(t) + B_c u(t) \\ y(t) & = & C_c x(t) + D_c u(t) \end{array} , \tag{2}$$

where $x(t) \in \mathcal{R}^{n_x}$ is the *state vector*, $u(t) \in \mathcal{R}^m$ the *input vector*, $y(t) \in \mathcal{R}^p$ the *output vector* and $A_c \in \mathcal{R}^{n_x \times n_x}$, $B_c \in \mathcal{R}^{n_x \times m}$, $C_c \in \mathcal{R}^{p \times n_x}$ and $D_c \in \mathcal{R}^{p \times m}$ are the model matrices. Often $D_c = 0$. As it is well known, this model class is adequate for a large number of real situations.

Now, we discretize in time this model. The discretization can be made according to any of the existing good numerical methods (see e.g. [17] [18]). Let T_{sc} be the time-discretization step, $x(k) \approx x_c(k\, T_{sc})$ be the state vector of the discrete-time model, and consider for simplicity the well-known θ-method [19]: it approximates the time-derivative with a difference quotient and substitutes the right-hand-side with a weighted average of its values at the time-instants k and $k+1$, depending on the parameter $0 \leq \theta \leq 1$:

$$\frac{x(k+1) - x(k)}{T_{sc}} = (1-\theta) f(x(k), u(k)) + \theta f(x(k+1), u(k+1)) \quad . \tag{3}$$

In this way, the time derivatives of the state variables are explicitly approximated in the discrete model. Using (3) with e.g. $\theta = 1$ (the Implicit Euler method) we obtain from (2) a state-space discrete model in physical coordinates:

$$\begin{array}{l} x(k+1) = A_f x(k) + B_f u(k) \\ y(k) = C_f x(k) \end{array} \tag{4}$$

with

$$A_f = (I - T_{sc} A_c)^{-1} \quad , \; B_f = (I - T_{sc} A_c)^{-1} T_{sc} B_c \quad , \; C_f = C_c \tag{5}$$

This is an example of a common nonlinear relation between the physical parameters of the continuous-time system matrices A_c, B_c and the discrete-time model matrices in physical coordinates A_f, B_f, that will be relevant in the work here presented. It is easy to demonstrate that if A_c is diagonalizable, i.e. there exists an invertible matrix U such that $U^{-1} A_c U = \Lambda$, then A_f is also diagonalizable and it holds:

$$U^{-1} A_f U = (I - T_{sc} \Lambda)^{-1} \quad . \tag{6}$$

In this paper, we refer to the problem of finding the minimal realization in the physical base of the system model, i.e. (4), starting from the minimal realization found by subspace methods in a data-driven base. Note that, since both realizations describe the same input-output data, they must be related by a change of basis. Indeed, as a general rule, let $\{A, B, C\}$ the matrices of a DLTI system and let T a basis-change matrix, then the transformed system is

$$\tilde{A} = T^{-1}AT, \quad \tilde{B} = T^{-1}B, \quad \tilde{C} = CT, \quad \tilde{x} = T^{-1}x \tag{7}$$

and the inputs $u(k)$ and the outputs $y(k)$ remain unchanged, as is well known from system's theory (and can be easily verified). This transformation has many other invariants, see Appendix A, and especially the eigenvalues of the matrix A, as can be directly seen from (7).

Now, to be more precise, let us call $\{A_s, B_s, C_s\}$ the matrices of the minimal realization found by subspace methods, and T_f the (unknown) basis-change matrix that transforms this data-driven minimal realization to the minimal realization in the physical base (4). We assume to know the true (physical) C_f matrix, because it can be easily built from the definition of the state-vector in the physical-mathematical model of the real system and the definition of the system outputs y. However, unfortunately, the matrix $T_f \in R^{n_x \times n_x}$ cannot be computed by simply using (7), since the linear system with multiple right-hand-side

$$C_s T_f = C_f \tag{8}$$

is underdetermined when, as we suppose, the number n_x of state variables is bigger than the number p of outputs, i.e. the output matrices C_f and C_s turn out to be rectangular, with dimensions $p \times n_x$.

The aim of this paper is to find a basis-change matrix $\hat{T}_f$ that approximates T_f in the sense that it transforms the minimal realization found by subspace methods to a base which is at least partially physical. This will be done by adding equations to (8) to get a well determined system, as it will be described in sec. 3.1.

Then, we will able to estimate at least some of the parameters of the continuous-time model (2), using the approximated minimal realization in the physical base and the inversion of the nonlinear map (5). Note that we are interested in a minimal realization in the physical base, since only in this basis it holds the relation (5) between the discrete model matrices and the physical parameters defined in the continuous model matrices.

3. Analysis and algorithm formulation

Let us start the analysis of the problem by assuming that A_c is diagonalizable and, by (6), it is also A_f. In sec. 4 we will generalize to non-diagonalizable matrices. Moreover, we suppose that the eigenvalues of A_f are well estimated by the eigenvalues of A_s,

where $\{A_s, B_s, C_s\}$ are the matrices of the minimal realization found by the subspace methods. Hence, we assume that A_s is equal to a similarity transformation of A_f (5) with an unknown basis-change matrix T_f, that transforms $A = A_s$ in $\bar{A} = A_f$, $B = B_s$ in $\bar{B} = B_f$ and $C = C_s$ in $\bar{C} = C_f$ according to (7). Therefore A_s is itself diagonalizable:

$$A_s = V_s \Lambda_{A_s} V_s^{-1} \tag{9}$$

where the components of the diagonal matrix Λ_{A_s} approximate well the eigenvalues of A_f, while the columns of V_s do not approximate well in general the eigenvectors of A_f. To describe this error would involve all the steps in the subspace methods and in the literature there is not much help in this direction. We will follow another path, instead. In this section we will define an algorithm to build an approximation of the matrix T_f, in the sense precised in the previous section, starting from two fundamental observations.

The first one is that the ordering of the eigenvectors is not an invariant for the minimal realization in the physical base. This can be easily proved noting that a permutation of the eigenvectors of A_f would consist in a basis-change which would change also the C_f matrix, which instead is fixed by the definition of the state variables and of the output variables. From the other side, there is no way to guarantee that the subspace methods find a minimal realization in which the eigenvectors of A_s are in the same order as those of A_f, since here C_s is completely arbitrary and data-driven.

This suggests us that there is an unknown, optimal permutation that should be applied to the eigenvectors of A_s, and we insert the search for this optimal permutation in our algorithm, as follows. First of all, by using V_s as a first basis change, we obtain a realization in modal coordinates

$$\{\Lambda_{A_s}, B_{V_s}, C_{V_s}\} \tag{10}$$

and get two advantages:

- Λ_{A_s} is a good estimate of A_f in modal coordinates, i.e. diagonalized. Indeed, only the diagonal elements of both matrices are different from zero and are equal to the eigenvalues, that we have supposed to be well estimated.
- the eigenspaces are now associated to single state variables and it is possible, with a row-column permutation, to associate them to specific measured variables, since the dynamics are decoupled, in this basis. To consider all the possible permutations for low-order models requires a modest effort and, as we will see, it turns out to be very effective on obtaining a good approximation of T_f among the infinite possible T.

The second fundamental observation regards the fact that we suppose to know the C_c $(= C_f)$ matrix, as already said in the previous section. Now, we can use it to bring the system at least partially in physical coordinates, through another change of coordinates, as we will see in sec. 3.1

Therefore, to build the algorithm we start from the observation that it exists a unique modal representation, except for a possible permutation of the eigenvalues on the diagonal. From each permutation we can bring the system at least partially in physical coordinates using our knowledge of the C matrix. Then, in sec. 3.2 we will see how to choose the optimal permutation.

All things considered, the proposed algorithm is the following Algorithm 1.

Algorithm 1 "Minimal realization in physical base"

1: given a set of I/O data, find a minimal realization $\{A_s, B_s, C_s\}$ through a subspace algorithm;

2: diagonalize A_s and get the decomposition (9);

3: for each possible permutation of the eigenvalues/eigenvectors, use the permuted eigenvectors V_s to change the basis of the state vector in modal coordinates (10) and compute the basis-change matrix $\hat{T}_f$ defined in sec. 3.1;

4: find the optimal permutation as indicated in sec. 3.2 and compute the resulting minimal realization in partial physical coordinates.

3.1. Imposing the physical C matrix

Here we find a transformation $\hat{T}_f$ that brings in physical coordinates the part of the model better observed by the measurements. Let us suppose for simplicity that the state variables are divided into two subsets $x = \begin{bmatrix} x[i_m] \\ x[i_u] \end{bmatrix}$, measured state variables first and then unmeasured ones, i.e. the matrix C_f is composed by the first p rows of the n_x-th order identity matrix, and that matrices $A_c/A_f/A_s$ and $B_c/B_f/B_s$ be partitioned accordingly:

$$A = \begin{bmatrix} A[i_m,:] \\ A[i_u,:] \end{bmatrix} \ , \ B = \begin{bmatrix} B[i_m,:] \\ B[i_u,:] \end{bmatrix} \ . \tag{11}$$

Note that some real systems will fit exactly in this subdivision of the state variables, some other will fit less and the distinction between "measured" and "unmeasured" state variables will be less sharp: "measured" variables are those that influence more the output variables than the "unmeasured" ones.

Now, we compute a basis-change matrix T_x, obtained as the solution of the following system with multiple right-hand-side:

$$MT_x = G \ , \ M = \begin{bmatrix} C_{V_s} \\ X \end{bmatrix} \ , \ G = \begin{bmatrix} C_f \\ Y \end{bmatrix} \tag{12}$$

where $M, T_x, G \in \mathcal{R}^{n_x \times n_x}$. The matrices X and Y can be chosen in different ways, each one representing a different method, whose choice will be left as an option to the proposed algorithm. Let us define $H_r^{\perp}$ the matrix whose rows form a basis for the

orthogonal complement of the row space of C_{V_s}, and $I[i_u, :]$ the matrix formed by the rows of the identity matrix of indexes corresponding to the unmeasured state variables. We found a few reasonable choices/methods to choose X and Y:

1. $X = Y = H_r^{\perp}$
2. $X = Y = I[i_u, :]$
3. $X = H_r^{\perp}$, $Y = I[i_u, :]$
4. $X = Y = H_r^{\perp} V_{As}$
5. $X = H_r^{\perp} V_{As}$, $Y = I[i_u, :]$

The system (12) must be interpreted in the following way: each row of M is transformed by T_x in the corresponding row of G. The matrix T_x will transform in particular the rows of C_{Vs} to the a-priori known physical representation C_f. The overall transformation matrix,

$$\hat{T}_f = V_s T_x \tag{13}$$

approximates T_f in the sense that it gives a basis change which is at least partially physical. Note that, in particular, both $\hat{T}_f$ and T_f transform C_s in C_f, i.e.:

$$C_s \hat{T}_f = C_s T_f = C_f \tag{14}$$

3.1.1. Analysis of method 3: permutation invariance

Method 3 highlights the fact that there are choices of X and Y that give results that are invariant to permutations. Here we demonstrate this fact.

Lemma 1. *Consider problem (12), with the choice $X = H_r^{\perp}$ and $Y = I[i_u, :]$. If we change basis to the realization $\{\Lambda_{A_s}, B_{V_s}, C_{V_s}\}$ with $T^{-1} = M_r$ a row-permutation matrix, the absolute entries of matrix $\hat{A}_f$ are invariant.*

Proof. We consider the similarity transformation with $T^{-1} = M_r$, in this case the realization becomes:

$$\{M_r \Lambda_{A_s} M_r^T, M_r B_{V_s}, C_{V_s} M_r^T\} \tag{15}$$

Let $X = H_r^{\perp}$ be a matrix whose rows form a basis for the orthogonal complement of the row space of $C_{V_s} M_r^T$, and let $Y = I[i_u, :]$. Since C_f is composed by the first p rows of the n_x-th order identity matrix, the matrix G corresponds to the identity matrix, $M = \begin{bmatrix} C_{V_s} M_r^T \\ H_r^{\perp} \end{bmatrix}$ and, so, the solution of problem (12) is $T_x = M^{-1}$.

Following Algorithm 1,

$$\hat{A}_f = M M_r \Lambda_{A_s} (M M_r)^{-1} \tag{16}$$

We note that if $U\Sigma V^T$ is a SVD decomposition of C_{V_s}, then $C_{V_s}M_r^T = U\Sigma V^T M_r^T$. Hence, $H_r^\perp$ is equal to the last i_u columns of V, transposed and permuted.

The matrix M can be rewritten as:

$$M = \begin{bmatrix} C_{V_s}M_r^T \\ V[:, i_u]^T M_r^T \end{bmatrix} = \begin{bmatrix} C_{V_s} \\ V[:, i_u]^T \end{bmatrix} M_r^T \tag{17}$$

Therefore,

$$\hat{A}_f = \begin{bmatrix} C_{V_s} \\ V[:, i_u]^T \end{bmatrix} M_r^T M_r \Lambda_{A_s} M_r^T M_r \begin{bmatrix} C_{V_s} \\ V[:, i_u]^T \end{bmatrix}^{-1} = \begin{bmatrix} C_{V_s} \\ V[:, i_u]^T \end{bmatrix} \Lambda_{A_s} \begin{bmatrix} C_{V_s} \\ V[:, i_u]^T \end{bmatrix}^{-1} \tag{18}$$

The permutation matrix disappears in the formula, then the absolute entries of the matrix $\hat{A}_f$ are invariant. □

3.1.2. Analysis of method 2: eigenvectors permutation

The choice $X = Y = I[i_u, :]$, instead, produces a result that depends significantly on the permutations. However, we can show that it is sufficient to take only the permutations referred to the measured variables i_m. In fact, as we will see in sec. 3.2, we will consider only the estimated $\hat{A}_c[i_m, i_m]$ submatrix to choose the optimal permutation.

Lemma 2. *Consider problem (12), with the choice $X = Y = I[i_u, :]$. If we change basis to the realization $\{\Lambda_{A_s}, B_{V_s}, C_{V_s}\}$ with $T^{-1} = M_r$ a row-permutation matrix, we determine a specific selection of eigenvectors to form the matrix $\hat{T}_f^{-1}$. Moreover, the number of permutations to be considered is restricted to $\binom{n_x}{p}$.*

Proof. The realization, with the similarity transformation $T^{-1} = M_r$, becomes:

$$\{M_r\Lambda_{A_s}M_r^T, M_r B_{V_s}, C_{V_s}M_r^T\} \tag{19}$$

With the choice $X = Y = I[i_u, :]$, since C_f is composed by the first p rows of the n_x-th order identity matrix, the matrix G corresponds to the identity matrix, $M = \begin{bmatrix} C_{V_s}M_r^T \\ I[i_u, :] \end{bmatrix}$ and, so, the solution of problem (12) is $T_x = M^{-1}$.

Following Algorithm 1,

$$\hat{A}_f = M M_r \Lambda_{A_s} M_r^T M^{-1}$$

then

$$\hat{A}_c = \frac{1}{T_{sc}}(I_{n_x} - \hat{A}_f^{-1}) = \frac{1}{T_{sc}}(I_{n_x} - T_x^{-1} M_r \Lambda_{A_s}^{-1} M_r^T T_x) \tag{20}$$

where I_{n_x} is the identity matrix of dimension $n_x \times n_x$. We partition the matrices T_x^{-1} and $M_r \Lambda_{A_s}^{-1} M_r^T$ in the following way:

$$T_x^{-1} = \begin{bmatrix} C_{meas} & * \\ 0 & I \end{bmatrix} \quad T_x = \begin{bmatrix} C_{meas}^{-1} & * \\ 0 & I \end{bmatrix} \quad M_r \Lambda_{A_s}^{-1} M_r^T = \begin{bmatrix} \Lambda_{meas}^{-1} & 0 \\ 0 & * \end{bmatrix}$$

where the matrix C_{meas} corresponds to the block $[i_m] \times [i_m]$ of the matrix $C_{V_s} M_r^T$, Λ_{meas}^{-1} is the $[i_m] \times [i_m]$ block of the matrix $M_r \Lambda_{A_s}^{-1} M_r^T$, and I is the identity matrix of dimension $(n_x - p) \times (n_x - p)$. Hence, (20) becomes:

$$\hat{A}_c = \frac{1}{T_{sc}} \left(I_{n_x} - \begin{bmatrix} C_{meas} \Lambda_{meas}^{-1} C_{meas}^{-1} & * \\ 0 & * \end{bmatrix} \right)$$

Since we consider only the submatrix $[i_m] \times [i_m]$ of $\hat{A}_c$, we can only take the permutations M_r referred to the p measured state variables $x[i_m]$. In this way, the number of permutations is $\frac{n_x!}{(n_x-p)!}$.

$$\begin{aligned} \hat{A}_f &= M M_r \Lambda_{A_s} M_r^T M^{-1} = M M_r V_{A_s}^{-1} A_s V_{A_s} M_r^T M^{-1} \\ &= \begin{bmatrix} C_s V_{A_s} M_r^T \\ I[i_u, :] \end{bmatrix} M_r V_{A_s}^{-1} A_s V_{A_s} M_r^T \begin{bmatrix} C_s V_{A_s} M_r^T \\ I[i_u, :] \end{bmatrix}^{-1} \\ &= \begin{bmatrix} C_s \\ I[i_u, :] M_r V_{A_s}^{-1} \end{bmatrix} A_s \begin{bmatrix} C_s \\ I[i_u, :] M_r V_{A_s}^{-1} \end{bmatrix}^{-1} \end{aligned} \tag{21}$$

So, the matrix $\hat{T}_f^{-1} = \begin{bmatrix} C_s \\ I[i_u, :] M_r V_{A_s}^{-1} \end{bmatrix}$ depends on a specific selection of eigenvectors.

From the structure of the matrix $\hat{T}_f^{-1}$, we notice that the permutations refer to the last i_u rows of matrix $V_{A_s}^{-1}$. In this way, the number of permutations to be considered is restricted to $\frac{n_x!}{(n_x-p)!p!} = \binom{n_x}{p}$. □

Note that, if V_{A_s} is orthogonal, the matrix M_r exchanges the eigenvectors, so the last $n_x - p$ components of the diagonal of the matrix $\hat{A}_f$ are the corresponding eigenvalues.

3.2. Choice of the optimal permutation

By restricting our search to a few permutations, from potentially infinite basis change candidates, we got the possibility to build a viable algorithm. Now we should choose a unique, optimal permutation. Here a fundamental problem is that the algorithm cannot directly measure the efficacy of a given permutation in the estimation of parameters, since they are unknown and there is no evident algebraic characterization of the physical basis in the discrete model.

Therefore, let us introduce an additional information that can help us at distinguish the representation in the physical base from the others, non physical. Let us see two possible approaches, reasonable for applications.

In the first approach we suppose to know a coarse estimate of (at least) a few parameters of the continuous model, that we want to estimate more precisely with the algorithm proposed. With this novel piece of information we apply then an heuristic method, which we will validate with the numerical experiments of sec. 5. An example of a reasonable heuristic method may be e.g. the following:

1. from the coarse initial estimate of the parameters of the continuous model, compute the matrix $\tilde{A}_c$;
2. from $\hat{T}_f$ obtain $\hat{A}_f$ and compute $\hat{A}_c$ from (5); choose the permutation where the submatrix $\hat{A}_c[i_m, i_m]$ is closer, in a chosen norm, to $\tilde{A}_c[i_m, i_m]$.

Indeed, the row/column permutations of Λ_{A_s} and the corresponding transformations in partial physical coordinates, produce nonlinear variations of high magnitude to the $\hat{A}_c$ matrix entries, as we have seen in the experiments. This helps a lot to distinguish the optimal permutation even starting from a matrix discretized with a coarse estimate of the physical parameters, as confirmed in the experiments of sec. 5.

In the second approach, that for brevity we don't test in the numerical experiments, we suppose to know some performance indexes of the unmeasured state variables (e.g. the maximum value, the range of variation, etc.), and we choose the permutation that gives an estimated system whose unmeasured state variables fit at best these indexes. There are many possible indexes and this approach becomes too much application dependent, so its performances; for this reason it is not tested here at this level of generality.

3.3. Parameters estimation in the continuum model

We have seen that, from the obtained minimal realization $\{\hat{A}_f, \hat{B}_f, \hat{C}_f\}$ we can reconstruct, at least partially, the matrices A_c and B_c from (5) and therefore some physical parameters of the continuous-time model. The main problem with this operation is that it involves an inversion of the matrix $\hat{A}_f$ and this blends the values of its p-order submatrix, that likely well approximates the corresponding submatrix of A_f, with its other values that may be very far from the corresponding values of A_f.

Let us concentrate on the sub-matrices of A_c and B_c relative to the measured variables. From (5) and the definition of T_f, we have:

$$B_c = \frac{1}{T_{sc}}(I - I + {A_f}^{-1})B_f = \frac{1}{T_{sc}}({T_f}^{-1}{A_s}^{-1}T_f){T_f}^{-1}B_s$$

and, therefore:

$$C_f B_c = \frac{1}{T_{sc}} C_s {A_s}^{-1} B_s \tag{22}$$

which is computable exactly from the matrices estimated by the subspace methods. Similarly, for the components of the matrix A_c, we have:

$$C_f A_c = \frac{1}{T_{sc}}(C_f - C_f {T_f}^{-1} {A_s}^{-1} T_f) = \frac{1}{T_{sc}}(C_f - C_s {A_s}^{-1} T_f) \tag{23}$$

and:

$$C_f A_c {C_f}^T = \frac{1}{T_{sc}}(C_f {C_f}^T - C_s {A_s}^{-1} T_f {C_f}^T) \tag{24}$$

which is not directly computable, since T_f is unknown, but can be approximated by substituting $\hat{T}_f$ (13) to T_f.

Therefore, the submatrix of A_c related to the measured variables cannot be computed exactly, but only estimated. In the numerical experiments we will see that the particular structure of A_c and B_c, that can vary substantially through applications, can influence considerably the estimation accuracy of the parameters in the continuum model.

4. Extension to non-diagonalizable matrices

Let A_c be the model matrix, and we suppose now that it is not diagonalizable. In this case, in principle we could use the Jordan canonical form and repeat the same arguments of sec. 3 with a block-diagonal matrix instead of a diagonal matrix Λ_{A_s}, with little modifications. Alternatively, we can use the real Schur decomposition, where we get a triangular matrix, more difficult to treat, as we will see, but much more safe to obtain numerically.

Then, we apply the real Schur decomposition $A_c = QUQ^T$, where Q is a real unitary matrix and U is a real upper triangular matrix with blocks of order 1 and 2 on its diagonal. The eigenvalues of A_c are the elements of the diagonal blocks of U of order 1 and the eigenvalues of the diagonal blocks of U of order 2.

As before, we assume that the eigenvalues of A_f are well estimated by the eigenvalues of A_s, where $\{A_s; B_s; C_s\}$ are the matrices of the minimal realization found by the subspace methods. We compute the Schur decomposition $A_s = Q_s U_s Q_s^T$, and, by using Q_s as a first basis change, we obtain the realization

$$\{U_s; B_{Q_s}; C_{Q_s}\}$$

It can be easily proved that Lemma 1 continues to hold, by substituting Λ_{A_s} with the real upper triangular matrix U_s. In the context of Lemma 2, we partition the matrices T_x^{-1} and $M_r U_s^{-1} M_r^T$ in the following way:

$$T_x^{-1} = \begin{bmatrix} C_{meas} & C_{nonmeas} \\ 0 & I \end{bmatrix} \quad T_x = \begin{bmatrix} C_{meas}^{-1} & -C_{meas}^{-1} C_{nonmeas} \\ 0 & I \end{bmatrix}$$

$$M_r U_s^{-1} M_r^T = \begin{bmatrix} U_1 & U_2 \\ U_3 & U_4 \end{bmatrix}$$

where the matrix C_{meas} corresponds to the block $[i_m] \times [i_m]$ of $C_{Q_s} M_r^T$ and C_{nomeas} corresponds to the block $[i_m] \times [i_u]$ of $C_{Q_s} M_r^T$ again. We note that, in general, the matrix $M_r U_s^{-1} M_r^T$ may not be upper triangular, i.e. $U_3 \neq 0$. Hence, (20) becomes:

$$\begin{aligned} \hat{A}_c &= \frac{1}{T_{sc}} (I_{n_x} - T_x^{-1} M_r U_s^{-1} M_r^T T_x) \\ &= \frac{1}{T_{sc}} \left(I_{n_x} - \begin{bmatrix} C_{meas} U_1 C_{meas}^{-1} + C_{nomeas} U_3 C_{meas}^{-1} & * \\ * & * \end{bmatrix} \right) \end{aligned} \tag{25}$$

thus showing a dependence of $A_c[i_m, i_m]$ from C_{nomeas} and, therefore, from the unmeasured state variables $x[i_u]$ and their permutations. Therefore, in principle we can not consider only the permutations referred to the measured variables. Actually, U_3 can contain only some off-diagonal terms of $U_s^{-1}[i_m, i_m]$ ($U_3 = M_r[i_u, i_m] U_s^{-1}[i_m, i_m] M_r[i_m, i_m]^T +$ other terms) and therefore the contribution of $C_{nomeas} U_3 C_{meas}^{-1}$ to the entries of $\hat{A}_c$ is much lower than that of $C_{meas} U_1 C_{meas}^{-1}$ ($U_1 = M_r[i_m, i_m] U_s^{-1}[i_m, i_m] M_r[i_m, i_m]^T +$ other terms). It means that neglecting this dependence on the permutations of unmeasured state variables may be often acceptable, as confirmed by the numerical experiments of sec. 5.

However, using exact arguments, we have:

$$\hat{A}_f = M M_r U_s M_r^T M^{-1} = \begin{bmatrix} C_s \\ I[i_u, :] M_r Q_s^{-1} \end{bmatrix} A_s \begin{bmatrix} C_s \\ I[i_u, :] M_r Q_s^{-1} \end{bmatrix}^{-1} \tag{26}$$

where the permutations refer to the last i_u rows of matrix Q_s^{-1} and then the number of permutations to be considered is restricted anyway to $\frac{n_x!}{p!}$.

5. Results

In this section we show the results of some relevant numerical experiments to assess the ability of Algorithm 1, and in particular of the basis change $\hat{T}_f$ (13) with an optimal permutation (see sec. 3.2), at estimating a reasonable approximation of the minimal realization in the physical base. More precisely, we will measure the relative estimation error for the parameters $A_c[i_m, i_m]$ of the continuous model (2), i.e. the components of A_c corresponding to the measured state variables $x[i_m]$. We will concentrate on diagonal entries of A_c, since in this general settings are the most indicative. Therefore, let us define the (vector) parameter estimation error indicator for the i-th experiment:

$$E_i = |diag(\hat{A}_c[i_m, i_m] - A_c[i_m, i_m] / A_c[i_m, i_m])| \tag{27}$$

where the division "/" and the modulus "$|\cdot|$" operators are meant componentwise.

Several subspace methods can be used to find a minimal realization, for example, MOESP [20], N4SID [21], CVA [22] and the algorithm here presented is independent from

the subspace method adopted. In our experiments, we apply the simplest variant of the subspace methods, where a straightforward singular values decomposition of the input-output data-matrix $G(\mathbf{y})$ is used to estimate the observability matrix (see Appendix B).

The algorithm has been tested with input-output data coming from two distinct model classes:

1. a sequence of models obtained from random change of basis, starting from a deterministic diagonal matrix Λ_{A_c}, i.e. the matrix A_c is obtained as $A_c = T^{-1}\Lambda_{A_c}T$. Various choices for T are considered (see the following subsections). In this way we can measure the ability of the estimation algorithm to tackle with an arbitrary realization, under the same dynamics. We tested systems of order n_x, varying from 3 to 12. For each n_x value in this range, the entries of Λ_{A_c} (the eigenvalues) are n_x entries of this vector: $[-0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.15, -0.1, -0.05, -0.01]$, taken at random positions. When we want a model with a non-diagonalizable matrix A_c, we add to Λ_{A_c} an upper-triangular matrix $A_{offdiag}$, with zeros on the main diagonal and uniformly random numbers in the upper off-diagonal entries, multiplied by a constant $c_{offdiag}$, i.e.

$$A_c = T^{-1}\left(\Lambda_{A_c} + c_{offdiag}\ A_{offdiag}\right)T \tag{28}$$

2. a sequence of mass-spring-damper systems, i.e. a linear elasto-dynamical model of the kind:

$$M\ddot{d}(t) + G\dot{d}(t) + Kd(t) = f(t) \tag{29}$$

with a variable n_x and matrices of physical parameters M, C and K taken as random perturbations from a fixed configuration of masses $m_i = 27.0$, dampers $c_i = 1.0e4$ and springs $k_i = 1.0e5$. Here, M, C and K are respectively the mass, damping and stiffness matrices, and $f(t)$ is the applied force. As is well known, defining the state vector $x(t) = \begin{bmatrix} \dot{d}(t) \\ d(t) \end{bmatrix}$ and discretizing in time one obtains a state-space system in the form (1). In these experiments, the outputs are supposed to be the measurements of the first p state variables. We tested systems of order n_x, varying from 4 to 100.

These two model classes have been chosen by following this consideration: the first one is sensibly random, i.e. where A_c has non-smooth eigenvectors. The second one is instead the discretization of a smooth operator (linear elastodynamics). See in Fig. 1 a snapshot of one sample eigenvector for each of the two cases. Many other benchmark examples can be found in the well known library SLICOT, see [23].

Each experiment consists of $N = 100$ estimation runs. In each run, there is one piecewise-constant input acting on a random B_c matrix. The number of outputs p varies

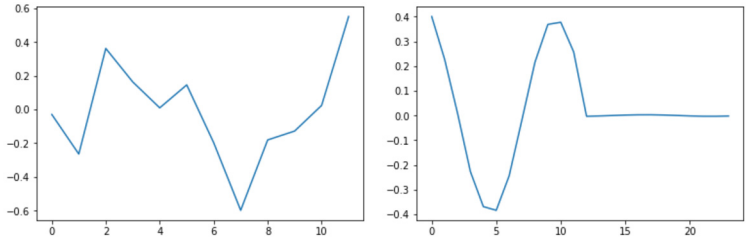

Fig. 1. Left: a sample eigenvector for model class 1 with $n_x = 12$ and (right) a sample eigenvector for model class 2 with $n_x = 24$. In the x axis there are the adimensional vector components indexes, in the y the vector components values.

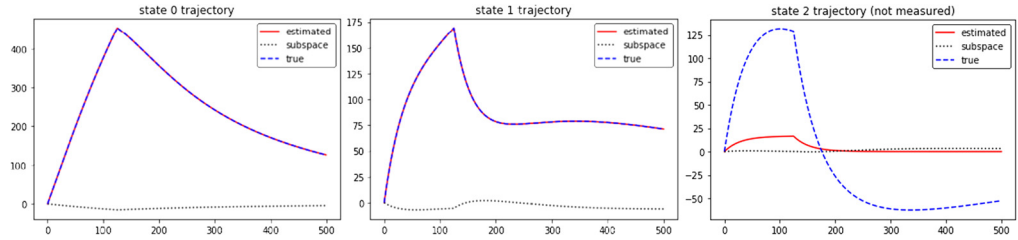

Fig. 2. Comparison of some state variables as a response to a piecewise constant input, of the "true" system, of the "subspace" system estimated by a standard subspace method and of the "estimated" system by Algorithm 1. Left: the time-behavior of state variable 0 (measured). Center: the time-behavior of state variable 1 (measured). Right: the time-behavior of state variable 2 (not measured). In the x axis there is the discrete time index.

from 1 to n_x and the matrix $C_c \in R^{p \times n_x}$ is simply the first p rows of the identity matrix of order n_x. To give an idea, in Fig. 2 we see a typical state-trajectory resulting from a piecewise-constant input; here $p = 2$ and state variables 0 and 1 are measured. Note that the "estimated" variables 0 and 1, i.e. obtained from the system estimated by Algorithm 1, are almost overlapping to the "true" ones, while the state-variables of the system "subspace", estimated by a standard subspace method, are quite far from the true ones. Note also that unmeasured state variables are not recovered to the physical base.

The results shown have been obtained by Algorithm 1 with the choice 2 of matrices X and Y, see section 3.1, since this is the choice we suggest.

These results have been obtained with a Python code developed by the authors, using NumPy for linear algebra computations. The code is available at the URL: https://github.com/NLALDlab/subspace-methods-in-physical-base.

5.1. Model class 1: case of diagonal A_c

First of all, let us consider the idealized case $T = I$, where we can measure some state variables from a minimal system with a diagonal A matrix (i.e. the system is expressed in modal coordinates). In this case, it is observed that this method reconstructs the

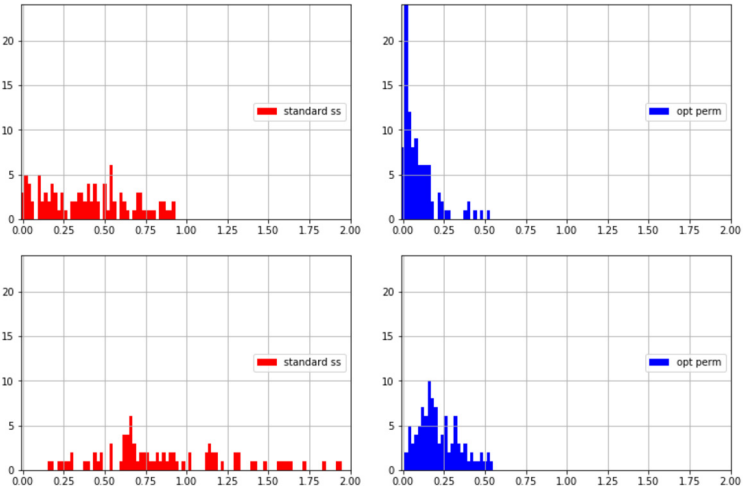

Fig. 3. Above left: histogram of $min(E_i),\ i = 0 \dots N-1$, with "standard ss"; above right: histogram of $min(E_i),\ i = 0 \dots N-1$ with "opt perm"; below left: histogram of $max(E_i),\ i = 0 \dots N-1$ with "standard ss"; below right: histogram of $max(E_i),\ i = 0 \dots N-1$ with "opt perm".

coefficients of $A_c[i_m, i_m]$ with a relative error close to the eigenvalues approximation error made by subspace methods, that is often very small.

5.2. Model class 1: case of diagonalizable A_c

In this case we suppose to collect data from a system where A_c is obtained by applying a generic orthogonal basis change T to the diagonal matrix Λ_{A_c}. The results are shown in Table 1. Note that the results with Algorithm 1 in ideal conditions ("opt_perm"), i.e. when the permutation is chosen by comparing with an initial estimate which is exact ($\tilde{A}_c = A_c$), show a substantial potential effectiveness of the approach, with respect to the plain subspace method ("standard ss") and also with respect to a mere imposition of the physical matrix C without permutations ("no perm"). An analogous effectiveness of the permutations has been revealed along all the experiments and, for brevity, the "no perm" results will not be reported in the following tables. In Table 1 are reported "mean" values of the estimation error E_i across N experiments, where the matrix T is varying randomly from experiment to experiment. There is a nontrivial dispersion around these "mean" values, i.e. for some T the results may be far from the mean accuracy attained. For this reason we showed here the median instead of the mean, which would be misleading. In Fig. 3 we see an example of the distribution of the results across N experiments. By analyzing the experiments individually, this dispersion cannot be described with general algebraic properties of T or the other matrices involved, like e.g. M of (12) and, therefore, should be ascribed to the nonlinear process of eigenvectors approximation made by the subspace method. This is outside the scope of this work and should be the subject of a further investigation involving subspace methods.

Table 1
The Table shows the results on the estimation error E_i (27), for various n_x and various estimation methods, from left to right: a standard subspace method ("standard ss"), $\hat{T}_f$ with no permutations ("no perm"), $\hat{T}_f$ with the best permutation obtainable by Algorithm 1 ("opt_perm"). Each cell contains two results: the median minimum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ and the median maximum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ through N experiments.

$p = 2$	$median\{min(E_i)\}_{i=0\dots N-1}, median\{max(E_i)\}_{i=0\dots N-1}$		
n_x	standard ss	$\hat{T}_f$ no perm	$\hat{T}_f$ opt perm
3	0.34, 0.90	0.62, 1.01	0.07, 0.24
4	0.35, 0.96	0.50, 1.80	0.11, 0.25
5	0.40, 1.08	0.61, 2.02	0.14, 0.28
6	0.42, 1.15	0.57, 1.77	0.10, 0.22
9	0.46, 1.10	0.73, 1.91	0.11, 0.21
12	0.44, 1.17	1.11, 1.79	0.12, 0.26

Table 2
The Table shows the results on the estimation error E_i (27), for various σ_{ini} and various estimation methods, from left to right: a standard subspace method ("standard ss") and with Algorithm 1 ("algo perm"). Each cell contains two results: the median minimum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ and the median maximum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ through N experiments.

$n_x = 6$, $p = 2$	$median\{min(E_i)\}_{i=0\dots N-1}, median\{max(E_i)\}_{i=0\dots N-1}$	
σ_{ini}	standard ss	$\hat{T}_f$ algo perm
0.0	0.33, 0.78	0.08, 0.19
0.1	0.37, 0.77	0.09, 0.21
0.2	0.33, 0.76	0.10, 0.26
0.3	0.36, 0.76	0.11, 0.29
0.4	0.39, 0.75	0.13, 0.33
0.5	0.30, 0.77	0.14, 0.34

The results of Table 1 are referred to the optimal permutation, that is obtained when the initial estimate is very close to the true parameter values, and in Table 2 we show how the results change with an increasing distance of the initial estimates from the true parameter values. This distance is characterized by the quantity σ_{ini} according to the following formula:

$$\tilde{A}_c = (1 + g)\, A_c \tag{30}$$

where $g \in \mathcal{N}(0, \sigma_{ini}^2)$, i.e. is a gaussian random variable with zero mean and variance σ_{ini}^2. In the following, we distinguish this case with the label "algo perm", while the case $\sigma_{ini} = 0$ is labeled with "opt_perm". Note that the values in the column "standard ss" oscillate around a constant value; indeed, the standard subspace method does not depend from an initial estimate of the true parameters. The column "algo perm", instead, show a moderate increase.

Note that, if we consider a basis change matrix T non orthogonal, we get slightly worse results, as reported in Table 3.

Last, in Table 4 we see the accuracy of the estimates as a function of the number of measured state variables.

Table 3
The Table shows the results on the estimation error E_i (27), for various combinations of n_x and various estimation methods, from left to right: a standard subspace method ("standard ss"), $\hat{T}_f$ with the best permutation obtainable by Algorithm 1 ("opt_perm"). Each cell contains two results: the median minimum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ and the median maximum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ through N experiments.

$p=2$	$median\{min(E_i)\}_{i=0\dots N-1}, median\{max(E_i)\}_{i=0\dots N-1}$	
n_x	standard ss	$\hat{T}_f$ opt perm
3	0.49, 2.18	0.19, 0.54
5	0.68, 3.28	0.29, 0.62
7	0.68, 3.19	0.28, 0.63

Table 4
The Table shows the results on the estimation error E_i (27), for various combinations of p and various estimation methods, from left to right: a standard subspace method ("standard ss") and the best permutation obtainable by Algorithm 1 ("opt_perm"). Each cell contains two results: the median minimum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ and the median maximum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ through N experiments.

$n_x=6$	$median\{min(E_i)\}_{i=0\dots N-1}, median\{max(E_i)\}_{i=0\dots N-1}$	
p	standard ss	$\hat{T}_f$ opt perm
1	0.63, 0.63	0.11, 0.11
2	0.42, 0.84	0.08, 0.20
4	0.11, 1.17	0.03 , 0.25
5	0.14, 1.07	0.01 , 0.34
6	0.05, 1.04	$1.e^{-12}$, $1.e^{-10}$

Table 5
The Table shows the results on the estimation error E_i (27), for various combinations of $c_{offdiag}$ and various estimation methods, from left to right: a standard subspace method ("standard ss"), the best permutation obtainable by Algorithm 1 ("opt_perm") and the best permutation obtainable by Algorithm 1 with $n_x!$ permutations ("opt_all-perm"). Each cell contains two results: the median minimum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ and the median maximum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ through N experiments.

$n_x=6$, $p=2$	$median\{min(E_i)\}_{i=0\dots N-1}, median\{max(E_i)\}_{i=0\dots N-1}$		
$c_{offdiag}$	standard ss	$\hat{T}_f$ opt perm	$\hat{T}_f$ opt all-perm
0.0	0.36, 0.83	0.13, 0.31	0.15, 0.32
0.1	0.35, 0.81	0.17, 0.38	0.11, 0.29
0.2	0.42, 0.87	0.14, 0.39	0.16, 0.39
0.3	0.32, 0.80	0.13, 0.34	0.15, 0.35
0.5	0.53, 1.19	0.21, 0.50	0.23, 0.48

5.3. Model class 1: case of non diagonalizable A_c

In this case we build A_c from a randomly generated real Schur decomposition (28). In Table 5 we see that the results are the same if consider only the $\frac{n_x!}{(n_x-p)!p!}$ permutations on the measured state-variables indexes i_m ("opt_perm"), that in this case are 15 permutations, and if we consider all $n_x!$ possible permutations ("opt_all-perm"), that in this case are 720 permutations, as was realized in sec. 4.

Table 6
The Table shows the results on the estimation error E_i (27), for various n_x and various estimation methods, from left to right: a standard subspace method ("standard ss") and $\hat{T}_f$ with the best permutation obtainable by Algorithm 1 ("opt_perm"). Each cell contains two results: the median minimum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ and the median maximum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ through N experiments.

$p = 2$ n_x	$median\{min(E_i)\}_{i=0\dots N-1}, median\{max(E_i)\}_{i=0\dots N-1}$	
	standard ss	$\hat{T}_f$ opt perm
4	0.86, 1.12	0.01, 0.03
6	0.94, 1.21	0.06, 0.13
10	0.87, 1.52	0.03, 0.09
20	0.97, 1.17	0.17, 0.45
30	1.00, 1.38	0.52, 0.80

5.4. Model class 2

In this case we suppose to collect data from a system where A_c represents the linear elasto-dynamical model (29). The results are shown in Table 6. Note that the results with Algorithm 1 are better than the model case 1, and are quite good up to $n_x \leq 20$. Note also that when increasing n_x the estimation error grows considerably. This opens up a certain amount of subtopics to be investigated, mainly related to eigenvectors approximation by subspace methods.

6. Conclusions

In this paper we have shown that, even when not all the state variables are measured, the model estimated by subspace methods can be partially transformed in physical coordinates and, therefore, useful to measure at least some of its physical parameters. The remaining part of the model, expressed in abstract, not physical coordinates, is actually black-box. Ultimately, this approach to physical parameters estimation produces in general a partial grey-box model.

The proposed algorithm allows a user choice, i.e. matrices X and Y of section 3.1. We suggest choice 2 mainly because it gives good results for all combinations of n_x and p (see sec. 5 for some examples) and it allows to substantially reduce the number of permutations to be considered, as stated in Lemma 2.

This algorithm can be used to estimate the physical parameters of a continuous-time model, with an arbitrary discretization method. Obviously, it means that these estimates are affected also by the space/time model discretization error and this should be in some way accounted for [24].

Note that when applying this method in practice, there are usually a-priori informations about the specific application, that can be exploited to obtain improved results than those here shown in this complete general settings. The estimates produced by this algorithm may be also the starting point for nonlinear estimation procedures, thus giving them a better chance to converge.

There are some lines of future investigation: the existence of an analytic criterion to choose the optimal permutation and to study the estimation error.

Declaration of competing interest

There is no competing interest.

Acknowledgements

This research was funded by the Project BIRD192932 of the University of Padova.

Appendix A. Invariants along a basis change

The invariants of transformation (7) are:

- the eigenvalues of the matrix A, in fact two similar matrices have the same characteristic polynomial, and so the same eigenvalues. Let $\tilde{A} = T^{-1}AT$, then

$$\begin{aligned} p_\lambda(\tilde{A}) &= det(\lambda I - \tilde{A}) = det(\lambda I - T^{-1}AT) = det(T^{-1}(\lambda I - A)T) \\ &= det(T^{-1})det(\lambda I - A)det(T) = p_\lambda(A); \end{aligned} \tag{A.1}$$

- the Markov coefficients $G_k = CA^{k-1}B$, in fact

$$G_k = CA^{k-1}B = \tilde{C}T^{-1}T\tilde{A}T^{-1}\dots T\tilde{A}T^{-1}T\tilde{B} = \tilde{C}\tilde{A}^{k-1}\tilde{B} = \tilde{G_k}; \tag{A.2}$$

- the stability of the system, in fact the eigenvalues are the same;
- the index of observability, indeed the system is observable if $rank(O_n) = rank(\begin{bmatrix} C \\ CA \\ \vdots \end{bmatrix}) = n$, then

$$O_n = \begin{bmatrix} \tilde{C}T^{-1} \\ \tilde{C}T^{-1}T\tilde{A}T^{-1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \tilde{C} \\ \tilde{C}\tilde{A} \\ \vdots \end{bmatrix} T^{-1} = \tilde{O_n}T^{-1} \tag{A.3}$$

  Then $rank(\tilde{O_n}) = rank(O_nT) = rank(T) = n$, because T is a non-singular matrix;
- the index of reachability, proof similar to the previous one;
- the energy of the system. Let $W_\infty = R_\infty R_\infty^T$ the reachability gramian, where $R_\infty = [B \quad AB \quad A^2B \quad \dots]$, so the energy is defined by

$$||u||_2^2 = x^T W_\infty^{-1} x \tag{A.4}$$

Under similarity transformation the gramian becomes:

$$R_\infty = [B \quad AB \quad \dots] = [T\tilde{B} \quad T\tilde{A}T^{-1}T\tilde{B} \quad \dots] = T[\tilde{B} \quad \tilde{A}\tilde{B} \quad \dots] \tag{A.5}$$

So $R_\infty = T\tilde{R_\infty} \rightarrow \tilde{R_\infty} = T^{-1}R_\infty \rightarrow \tilde{W_\infty} = \tilde{R_\infty}\tilde{R_\infty^T} = T^{-1}W_\infty(T^{-1})^T$
Hence: $W_\infty = T\tilde{W_\infty}T^T \rightarrow W_\infty^{-1} = (T^T)^{-1}\tilde{W_\infty}^{-1}T^{-1}$
In this way, the energy is

$$||u||_2^2 = x^T W_\infty^{-1} x = \tilde{x}^T T^T (T^T)^{-1}\tilde{W_\infty}^{-1}T^{-1}T\tilde{x} = \tilde{x}^T\tilde{W_\infty}^{-1}\tilde{x} \tag{A.6}$$

- the property of being diagonalizable, in fact if A is diagonalizable, then there exists an invertible matrix U such that $U^{-1}AU = \Lambda$. Now $\tilde{A} = T^{-1}AT$, so $U^{-1}T\tilde{A}T^{-1}U = \Lambda$. In this way $\tilde{A}$ is diagonalizable because the matrix $T^{-1}U$ is invertible.

Appendix B. Subspace system identification

Let us consider a general discrete, linear, time-invariant state-space model [4]:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k) + Du(k) + w(k) \end{aligned} \tag{B.1}$$

where $v(k)$ and $w(k)$ represent mutually and serially uncorrelated zero-mean error processes with covariance matrices R and Q, respectively. The model assumes a prior distribution for $x(0)$ with $E\{x(0)\} = \mu$ and $cov(x0) = \Sigma$. It is a standard assumption that the state $x(0)$ is taken to be uncorrelated with $v(k)$ and $w(k)$ for all k. We assume normality of both error processes as well as for $x(0)$. Given a sequence of measured inputs and outputs, subspace methods estimate the system matrices (A, B, C, D), the initial state vector $x(0)$ and the statistical description of the error processes, i.e. the covariance matrices R and Q or the Kalman gain K if model (B.1) is rewritten in innovation form [4].

Let us give an outline of these methods, for simplicity in the deterministic case $v(k) = w(k) = 0$. Let's start by considering the state at time instant k as a function of the initial state vector $x(0)$ and the input sequence $u(k)$:

$$x(k) = A^k x(0) + \sum_{i=0}^{k-1} A^{k-i-1}Bu(i)$$

from which, iteratively, one easily obtains:

$$\begin{bmatrix} y(0) \\ \vdots \\ y(r-1) \end{bmatrix} = \begin{bmatrix} C \\ \vdots \\ CA^{r-1} \end{bmatrix} x(0) + \begin{bmatrix} D & 0 & \dots & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ \vdots & & \ddots & & \\ CA^{r-2}B & \dots & \dots & CB & D \end{bmatrix} \begin{bmatrix} u(0) \\ \vdots \\ u(r-1) \end{bmatrix}$$

$$\triangleq O_r\ x(0) + S_r \begin{bmatrix} u(0) \\ \vdots \\ u(r-1) \end{bmatrix} \tag{B.2}$$

where r is some arbitrary positive integer: $n_x < r \ll N$, with n_x the model order of the system and N is the maximum system order conceivable, in the considered application. Usually it is assumed that the integer N can be chosen such that the matrix $U_{0,r,N}U_{0,r,N}^T$ has full row rank, where $U_{0,r,N}$ is $r \times N$ block Hankel matrix built from the sequence $\mathbf{u} = \{u(k)\}$, where the index of the first block element is 0. The starting point of subspace methods is the matrix equation

$$Y_{0,r,N} = O_r X_{0,1,N} + S_r U_{0,r,N} \tag{B.3}$$

where $Y_{0,r,N}$ is $r \times N$ block Hankel matrix constructed from the sequence $\mathbf{y} = \{y(k)\}$, where the index of the first block element is 0, the matrix $X_{0,1,N}$ has as its columns the states $x(k)$, $k = 0, .., N-1$ and the matrices O_r, called the extended observability matrix, and S_r contain all model parameters (A, B, C, D). Subspace methods first form the matrix

$$G(\mathbf{y}) = Y_{0,r,N}\Pi_U^{\perp} = O_r X_{0,1,N}\Pi_U^{\perp} \tag{B.4}$$

from (B.3), where $\Pi_U^{\perp} = I - U^T(UU^T)^{-1}U$ is the orthogonal projection matrix on the nullspace of $U := U_{0,r,N}$. Once the model order n_x is determined, a truncated SVD of G is used to estimate the matrix O_{r,n_x}, i.e. the matrix O_r with n_x columns; this decomposition is then used to compute a system realization and the initial state estimate. However, if the measured data are affected by noise, the equation (B.4) can be rewritten as $G(\mathbf{y}) = Y_{0,r,N}\Pi_U^{\perp} = O_r X_{0,1,N}\Pi_U^{\perp} + E\Pi_U^{\perp}$, where E contains the contribution of the noise.

References

[1] B. De Schutter, Minimal state-space realization in linear system theory: an overview, in: Special Issue on Numerical Analysis in the 20th Century – Vol. I: Approximation Theory, J. Comput. Appl. Math. 121 (1–2) (2000) 331–354, https://doi.org/10.1016/S0377-0427(00)00341-1.

[2] B. Ho, An effective construction of realizations from input/output descriptions, Ph.D. Thesis, Stanford University, Stanford, CA, 1966.

[3] B. Ho, R. Kalman, Effective construction of linear, state-variable models from input/output functions, Regelungstechnik 14 (12) (1966) 545–548.

[4] P. Van Overschee, B. De Moor, Subspace Methods in System Identification, Springer, 1996.

[5] Z. Liu, A. Hansson, L. Vandenberghe, Nuclear norm system identification with missing inputs and outputs, Syst. Control Lett. 62 (8) (2013) 605–612, https://doi.org/10.1016/j.sysconle.2013.04.005.

[6] M. Verhaegen, A. Hansson, N2sid: nuclear norm subspace identification of innovation models, Automatica 72 (2016) 57–63, https://doi.org/10.1016/j.automatica.2016.05.021.

[7] G. Deolmi, F. Marcuzzi, A parabolic inverse convection–diffusion–reaction problem solved using space–time localization and adaptivity, Appl. Math. Comput. 219 (16) (2013) 8435–8454, https://doi.org/10.1016/j.amc.2013.02.040.

[8] A. Beghi, F. Marcuzzi, P. Martin, F. Tinazzi, M. Zigliotto, Virtual prototyping of embedded control software in mechatronic systems: a case study, Mechatronics 43 (2017) 99–111, https://doi.org/10.1016/j.mechatronics.2017.03.004.

[9] L. Ljung, System Identification: Theory for the User, Prentice-Hall, 1999.
[10] P. Parrilo, L. Ljung, Initialization of physical parameter estimates, in: Proceedings of the IFAC Symposium on System Identification, Rotterdam, The Netherlands, 2003.
[11] M.S.A. Alenany, H. Shang, I. Ziedan, Improved subspace identification with prior information using constrained least squares, IET Control Theory Appl. 5 (13) (2011) 1568–1576, https://doi.org/10.1049/iet-cta.2010.0585.
[12] I. Markovsky, G. Mercére, Subspace identification with constraints on the impulse response, Int. J. Control 90 (8) (2016) 1728–1735, https://doi.org/10.1080/00207179.2016.1219922.
[13] G. Mercére, O. Prot, J. Ramos, Identification of parameterized gray-box state- space systems: from a black-box linear time-invariant representation to a structured one, IEEE Trans. Autom. Control 59 (2014) 2873–2885.
[14] C. Lyzell, M. Enqvist, L. Ljung, Handling certain structure information in subspace identification, in: Proceedings of the IFAC Symposium on System Identification, Sain Malo, France, 2009.
[15] R.B.H. Luc, M. De Angelis, R. Longman, Constructing second-order models of mechanical systems from identified state space realizations. Part I: theoretical discussions, J. Eng. Mech. 129 (2003) 477–488.
[16] B. Olofsson, et al., Continuous-time gray-box identification of mechanical systems using subspace-based identification methods, in: 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2014, pp. 327–333.
[17] A. Quarteroni, Numerical Models for Differential Problems, Springer-Verlag Mailand, 2014.
[18] L.N. Trefethen, A. Birkisson, T.A. Driscoll, Exploring ODEs, SIAM-Society for Industrial and Applied Mathematics, USA, 2017.
[19] A. Iserles, A First Course in the Numerical Analysis of Differential Equations, Cambridge University Press, 2009.
[20] M. Verhaegen, P. Dewilde, Subspace model identification part 1. The output-error state-space model identification class of algorithms, Int. J. Control 56 (5) (1992) 1187–1210, https://doi.org/10.1080/00207179208934363.
[21] P. Van Overschee, B. De Moor, N4sid: subspace algorithms for the identification of combined deterministic-stochastic systems, Automatica 30 (1) (1994) 75–93.
[22] W. Larimore, Canonical variate analysis in identification, filtering, and adaptive control, in: 29th IEEE Conference on Decision and Control, vol. 2, 1990, pp. 596–604.
[23] D. Kreßner, V. Mehrmann, T. Penzl, Ctdsx - a Collection of Benchmark Examples for State-Space Realizations of Continuous-Time Dynamical Systems, 1998.
[24] F. Marcuzzi, Linear estimation of physical parameters with subsampled and delayed data, J. Comput. Appl. Math. 331 (2018) 11–22, https://doi.org/10.1016/j.cam.2017.09.036.