A software package for system identification in the behavioral setting

Ivan Markovsky

Department ELEC
Vrije Universiteit Brussel
imarkovs@vub.ac.be

Abstract

An identification problem with no a priori separation of the variables into inputs and outputs and representation invariant approximation criterion is considered. The model class consists of linear time-invariant systems of bounded complexity and the approximation criterion is the minimum of a weighted 2-norm distance between the given time series and a time series that is consistent with the model. The problem is equivalent to and is solved as a mosaic-Hankel structured low-rank approximation problem. Software implementing the approach is developed and compared with other packages on benchmark problems. Additional nonstandard features of the software are specification of exact and missing variables, fixed initial conditions, and identification from multiple time series with different length. An extended version of the paper is a literate program, implementing the method and reproducing the presented results.

Keywords: system identification, model reduction, behavioral approach, missing data, mosaic-Hankel low-rank approximation, reproducible research, DAISY.

1 Introduction

One of the main criticisms to the behavioral approach in system theory and control [21, 15] is that it is not supported by numerical algorithms and software. Despite its conceptual and theoretic advantages, the behavioral approach has produced only a few engineering tools; most notably the identification methods of [17] and [9]. Compared to the abundance of algorithms and software in the classical setting, this scarcity of tools accounts to a large extent for the slow acceptance of the behavioral approach by engineers.

In this paper, we present a flexible method for system identification in the behavioral setting, which is an outgrow of the method in [9]. The method is implemented in a publicly available software package which is fast, easy to use, and has the following nonstandard features.

1. *Representation free problem formulation*

   Classical identification problems are invariably defined in terms of representations of the system—transfer function, input/state/output, convolution kernel, *etc*. A problem can be equivalently posed in frequency domain as well as time domain, using input/state/output, convolution, or any other representation. Different representations have different advantages, so that all of them should be considered as options in solving the problem. We view the model representation as an implementation detail of a particular solution method and define the problem in terms of the desired system behavior—set of trajectories of the system.

   Related to the use of a representation invariant problem formulation, is the assumption in classical system identification problems that an input/output partitioning of the variables is given and fixed. The choice of inputs and outputs is not unique [22, Section VIII]. Different input/output partitions, generically, lead to the same optimal model's behavior [10]. In nongeneric cases, however, the optimal model is not compatible with a specified input/output partition. If such a partition is chosen, the identification problem is ill-posed. In practice, even a well-posed problem with fixed input/output partition can be ill-conditioned. Using the representation free setting and thus leaving the choice of the input-output partition to the identification method has the potential advantage of leading to methods that are more general conceptually (the input/output partition is identified together with the model), easier to analyze theoretically (solution always exists), and are computationally more robust (ill-conditioning is avoided).

2. *Multiple time-series*

   Another nonstandard feature of the method presented in the paper is the use of multiple time series. Multiple time series occur in practice when the measurement experiment is repeated. In static estimation problems, a measurement experiment yields a q-dimensional vector observation, so that the only way to increase the amount of data is to repeat the experiment. In dynamic problems, an experiment yields a q-dimensional vector time series. More data can be collected in this case also by increasing the length of the time series (number of samples in time). Our framework unifies the two approaches of data collection—repeated experiments and longer experiment. The former approach is of interest, for example, in the case of autonomous dynamical systems, where asymptotic analysis in time is not applicable.

   The problem of using the data available in multiple time series for identification of a single model is often addressed by heuristic methods. The rigorous method of [18] is based on concatenation of the separate time series into one time series, adding transient responses at the times of transition from one trajectory to another. This reduces the problem of identification from N time series to the problem of identification from a single time series *and* estimation of N transient responses—both being classical problems. This approach is implemented in all methods available in the System Identification Toolbox of Matlab [4].

   The method presented in this paper uses a different approach. The time series parameterize separate block-Hankel matrices, with fixed row dimension but possibly different column dimensions. The block-Hankel matrices are appended to each other, forming a $1 \times N$ block matrix with block-Hankel elements. Such a matrix is called a mosaic block-Hankel matrix. The identification problem using the multiple data sets is then equivalent to low-rank approximation of the mosaic block-Hankel matrix. Our approach is a natural generalization of the classical approach for solving static estimation problems where the blocks are the individual (vector) observations and the mosaic block-Hankel matrix is the unstructured $q \times N$ matrix of the stacked next to each other observations.

3. *Exact and missing data*

   Due to malfunctioning of measurement devices, communication channels, or storing devices, the available data may have missing measurements at some time instances. When the missing data span over ℓ or more sequential time instances, where ℓ is the lag of the system, the problem reduces to the problem of identification from two independent time series. Also for problems with a given input/output partitioning, missing output variables can be estimated by existing methods. More general patterns of missing data, *e.g.*, non sequential missing variables and missing variables in arbitrary combination of variables, require new methods. The approach used in the paper, treats missing data by minimizing a weighted approximation criterion with zero weights associated to the missing values.

   The opposite of a missing data value is an exact data value. An example of how exact data occur in practice is data collection experiment for a system starting "at rest", *i.e.*, zero initial conditions. This prior knowledge can be used in the identification problem by imposing equality constraints, setting the approximation of the exact values to the given data values. The approach of imposing exact values used in this paper, is by minimizing a weighted approximation criterion with infinite weights associated to the exact values. Thus, exact and missing values are treated uniformly as extreme cases of noise corrupted observations when the noise has zero and infinite variance, respectively.

4. *Software implementation in a literature programming style*

   Scientific publications rarely include full implementation details of the methods described. A software implementation of the method (when available as open source) necessarily contains the full details about the method and is therefore the ultimate source of information. Even well documented code, however, may be difficult to read and may not correspond to a description in related scientific publication. The literature programming style [2] is a possible solution to this issue. A literate program includes the source code as an integral part of the document describing it. A literate program is designed to be read by a human being rather than a computer.

   An extended version of this paper is a literate program for the described software package. The code is split into chunks and the chunks are presented in a logical sequence, which happens to be different from the sequence in which the chunks appear in the computer executable programs. The code chunks provide the implementation

details for the algorithmic steps and, vice verse, the presentation in the paper serves as documentation of the code. We use the `noweb` [16] syntax for writing literate programs. The computer executable code as well as the human readable text are automatically extracted from the literate program by `noweb`, so that the code and the paper coexist in common source files.

The approach for solving the identification problem, used in the paper, is weighted structured low-rank approximation [5]. Rank deficient mosaic-Hankel matrices are in a one-to-one correspondence with trajectories of linear time-invariant systems of bounded complexity. Therefore, data approximation by a linear time-invariant system of bounded complexity is a mosaic-Hankel low-rank approximation problem. The behavioral approach and the low-rank approximation setting fit like a hand fits a glove. The system theoretic notion of a linear time-invariant model of bounded complexity corresponds to the linear algebra notion of a rank deficient Hankel matrix. The linear algebra setting provides algorithms and software for addressing problems in the system theoretic setting.

Element-wise weights allow approximation with an emphasis on some variables and/or some time instances. Zero and infinite weights are allowed. A zero weight ignores the corresponding data point in the approximation and an infinite weight forces the approximation of the corresponding data point to be equal to the given one. This allows us to take into account missing and exact data.

Once the identification problem is cast as a mosaic-Hankel structured low-rank approximation problem, it is solved by the efficient methods of [19]. A practical implementation of the methods in C++, using optimization algorithms from the GNU scientific library [1], is available [7]. The identification function presented in this paper is a wrapper function to the structured low-rank approximation solver of [7]. The software is available from:

`http://homepages.vub.ac.be/~imarkovs/slra/software.html`

The paper is organized as follows: Section 2 sets the notation and states the considered identification problem. Section 3 presents the solution approach and gives details about the implementation of the software. Specific identification problems—L_2-optimal model reduction, identification from multiple trajectories of different lengths, and identification with missing data—are presented in Sections 4–6, respectively. In Section 7, the performance of the identification package is tested on real-life and simulated data base from the database for system identification DAISY.

2 Problem formulation

Model class: linear time-invariant systems of bounded complexity

A discrete-time dynamical system $\mathscr{B}$ is a collection of trajectories — q-variables time-series $w : \mathbb{Z} \to \mathbb{R}^q$. The class of finite dimensional linear time-invariant systems with q variables and at most $\mathtt{m}$ inputs is denoted by $\mathscr{L}^q_{\mathtt{m}}$. A linear time-invariant system $\mathscr{B} \in \mathscr{L}^q_{\mathtt{m}}$ admits a representation by constant coefficients difference equation

$$\mathscr{B} = \mathscr{B}(R) := \{\, w \mid R_0 w + R_1 \sigma w + \cdots + R_\ell \sigma^\ell w = 0 \,\}, \tag{DE}$$

where σ is the shift operator

$$(\sigma w)(t) = w(t+1).$$

The minimal natural number ℓ, for which there exists an ℓth order difference equation representation for $\mathscr{B}$ is an important invariant of the system, called the *lag*. The number of inputs and the lag specify the complexity of the system in the sense that the dimension of the restriction of $\mathscr{B}$ to the interval $[1,T]$, where $T \geq \ell$, is bounded by $T\mathtt{m} + \ell(q-\mathtt{m})$. The subset of $\mathscr{L}^q_{\mathtt{m}}$ with lag at most ℓ is denoted by $\mathscr{L}^q_{\mathtt{m},\ell}$.

No a priori separation of the variables into inputs and output is made, however, the variables w can always be partitioned into inputs u (free variables) and outputs y (dependent variables) and the system can be represented in the input/state/output representation

$$\mathscr{B} = \mathscr{B}(A,B,C,D,\Pi) := \{\, w = \Pi(u,y) \mid \text{there is } x, \text{ such that } \sigma x = Ax + Bu,\ y = Cx + Du \,\}, \tag{I/S/O}$$

where Π is a permutation matrix. If $\Pi = I_q$, it will be skipped, *i.e.*, $\mathscr{B}(A,B,C,D) = \mathscr{B}(A,B,C,D,I)$.

The number of inputs $\mathtt{m}$, the number of outputs $\mathtt{p} = q - \mathtt{m}$, and the minimal state dimension $\mathtt{n}$ of an input/state/output representation of $\mathscr{B}$ are invariant of the representation and in particular of the input/output partitioning. The order $\mathtt{n}$

of a state-space representation of a linear time-invariant system $\mathscr{B} = \mathscr{B}(R)$ with lag ℓ and $\mathtt{p}$ outputs is $\mathtt{n} \leq \ell\mathtt{p}$. In the case when the block $P_\ell \in \mathbb{R}^{\mathtt{p}\times\mathtt{p}}$ of $R_\ell = \begin{bmatrix} Q_\ell & -P_\ell \end{bmatrix}$ is nonsingular, $\mathtt{n} = \ell\mathtt{p}$ and $w = (u, y)$ is a possible input/output partition, *i.e.*, Π can be chosen equal to I. We make this simplifying assumption in the rest of the paper. The class of systems with q variables and inputs, order, and lag bounded by, respectively $\mathtt{m}$, $\mathtt{n}$, and ℓ is denoted by $\mathscr{L}^{q,\mathtt{n}}_{\mathtt{m},\ell}$.

Approximation criterion: distance between data and model

We measure the *misfit* (lack of fit) between the data w_{d} and the model $\mathscr{B}$ by the orthogonal distance from w_{d} to $\mathscr{B}$

$$M(w_{\mathrm{d}}, \mathscr{B}) := \min_{\widehat{w}^1,\ldots,\widehat{w}^k \in \mathscr{B}} \sqrt{\sum_{k=1}^{N} \|w_{\mathrm{d}}^k - \widehat{w}^k\|_2^2}. \tag{M}$$

Intuitively, the misfit shows how much the model $\mathscr{B}$ fails to "explain" the data w_{d}.

Missing elements are marked by the symbol `NaN` and are excluded from the approximation criterion, *i.e.*,

$$\mathtt{NaN} - \widehat{w}_i(t) = 0, \qquad \text{for all} \quad \widehat{w}_i(t) \in \mathbb{R}.$$

The optimal approximate modeling problem considered aims to find a system $\widehat{\mathscr{B}}$ in the model class $\mathscr{L}^q_{\mathtt{m},\ell}$ that best fits the data according to the misfit criterion.

Given a set of time series

$$w_{\mathrm{d}} = \{w_{\mathrm{d}}^1, \ldots, w_{\mathrm{d}}^N\}, \quad \text{where} \quad w_{\mathrm{d}}^k = \big(w_{\mathrm{d}}^k(1), \ldots, w_{\mathrm{d}}^k(T_k)\big), \quad w_{\mathrm{d}}^k(t) \in \mathbb{R}^q,$$

and a complexity specification $(\mathtt{m}, \ell)$, find the system

$$\widehat{\mathscr{B}} := \arg\min_{\mathscr{B} \in \mathscr{L}^q_{\mathtt{m},\ell}} M(w_{\mathrm{d}}, \mathscr{B}). \tag{SYSID}$$

Special cases of (SYSID) are static data modeling ($\ell = 0$) and output-only or autonomous system identification ($\mathtt{m} = 0$). The solution approach, described next, leads to an algorithm that covers these special cases. In addition,

1. elements of the given time series w_{d} can be specified as "missing" by passing the symbol `NaN` for their value;
2. elements of the given time series w_{d} can be specified as "exact", in which case they appear unmodified in the approximation $\widehat{w}$;
3. the approximation $\widehat{w}$ can be constrained to be a trajectory of the model $\mathscr{B}$, generated under a priori fixed initial conditions w_{ini}, see [6], *i.e.*,
$$\begin{bmatrix} w_{\mathrm{ini}} \\ \widehat{w} \end{bmatrix} \in \mathscr{B}.$$
(Note that problem (SYSID) identifies the model without prior knowledge about the initial conditions, under which the data w_{d} is generated, *i.e.*, w_{ini} is a free variable.)

3 Solution approach

For a given set of trajectories w_{d} and a natural number ℓ, we define the mosaic-block-Hankel matrix (a $1 \times N$ block matrix with block-Hankel blocks)

$$\mathscr{H}_{\ell+1}(w_{\mathrm{d}}) := \begin{bmatrix} \mathscr{H}_{\ell+1}(w_{\mathrm{d}}^1) & \cdots & \mathscr{H}_{\ell+1}(w_{\mathrm{d}}^N) \end{bmatrix}, \quad \text{where} \quad \mathscr{H}_{\ell+1}(w_{\mathrm{d}}^k) := \begin{bmatrix} w_{\mathrm{d}}^k(1) & w_{\mathrm{d}}^k(2) & \cdots & w_{\mathrm{d}}^k(T-\ell) \\ w_{\mathrm{d}}^k(2) & w_{\mathrm{d}}^k(3) & \cdots & w_{\mathrm{d}}^k(T-\ell+1) \\ \vdots & \vdots & & \vdots \\ w_{\mathrm{d}}^k(\ell+1) & w_{\mathrm{d}}^k(\ell+2) & \cdots & w_{\mathrm{d}}^k(T) \end{bmatrix}.$$

The solution method is based on the following equivalence

$$\big(w_{\mathrm{d}}(1),\ldots,w_{\mathrm{d}}(T_k-\ell)\big) \in \mathscr{B} \in \mathscr{L}^{q,\mathtt{n}}_{\mathtt{m},\ell}, \quad \text{for } k=1,\ldots,N \qquad \Longleftrightarrow \qquad \operatorname{rank}\big(\mathscr{H}_{\ell+1}(w_{\mathrm{d}})\big) \leq (\ell+1)\mathtt{m}+\mathtt{n},$$

i.e., the time series w_{d}, possibly except for the last ℓ samples, are exact trajectories of a linear time-invariant model $\mathscr{B}$ with complexity bounded by $(\mathtt{m},\mathtt{n})$ if and only if the mosaic-block-Hankel matrix $\mathscr{H}_{\ell+1}(w_{\mathrm{d}})$ has rank bounded by

$$r = (\ell+1)\mathtt{m}+\mathtt{n}.$$

Therefore, the identification problem (SYSID) is equivalent to the structured low-rank approximation problem

$$\text{minimize} \quad \text{over } \widehat{w} \quad \|w_{\mathrm{d}}-\widehat{w}\|_{\ell_2}^2 \quad \text{subject to} \quad \operatorname{rank}\big(\mathscr{H}_{\ell+1}(\widehat{w})\big) \leq r. \tag{SLRA}$$

Note 1. Under the assumption that P_ℓ is full rank, the whole trajectories $w_{\mathrm{d}}^1,\ldots,w_{\mathrm{d}}^N$ are in the behavior of the model.

Problem (SLRA) is a classic problem that can be solved in different ways. The approach used in the paper is based on the kernel representation of the rank constraint

$$\operatorname{rank}\big(\mathscr{H}_{\ell+1}(\widehat{w})\big) \leq r \qquad \Longleftrightarrow \qquad R\mathscr{H}_{\ell+1}(\widehat{w}) = 0 \quad \text{and} \quad R^\top R = I_{(\ell+1)q-r}. \tag{KER}$$

The matrix R in the right-hand-side of (KER) is related to the parameters $R_0,R_1,\ldots,R_\ell$ of the difference equation representation (DE) of the exact model for $\widehat{w}$ as follows:

$$R = \begin{bmatrix} R_0 & R_1 & \cdots & R_\ell \end{bmatrix}, \qquad \text{where} \quad R_i \in \mathbb{R}^{\mathtt{p}\times q}.$$

Algorithms and software for mosaic-Hankel low-rank approximation are developed in [8, 19]. We use the software of [7], so that the software presented in this paper is an interface to the structured low-rank approximation solver for the purpose of linear time-invariant system identification, see Figure 1. On its own, the structured low-rank approximation solver computes a parameter $\widehat{R}$ of the difference equation representation of the optimal approximating system $\widehat{\mathscr{B}}$. The system identification function converts the parameter $\widehat{R}$ to the parameters $(\widehat{A},\widehat{B},\widehat{C},\widehat{D})$ of an input/state/output representation of $\widehat{\mathscr{B}}$ in order to facilitate the usage of the model by other analysis and synthesis tools. In addition, a system $\mathscr{B}_{\mathrm{ini}} \in \mathscr{L}^q_{\mathtt{m},\ell}$, specified by parameters $(A_{\mathrm{ini}},B_{\mathrm{ini}},C_{\mathrm{ini}},D_{\mathrm{ini}})$ can be used as an initial approximation for the optimization algorithm. The parameters $(A_{\mathrm{ini}},B_{\mathrm{ini}},C_{\mathrm{ini}},D_{\mathrm{ini}})$ are converted to the parameter R_{ini}, used by the structured low-rank approximation solver. Details about these conversions are given in Appendix B. Although the identification function has as external interface the (I/S/O) representation of the model, the internal computations are done via the parameter R of the difference equation representation.

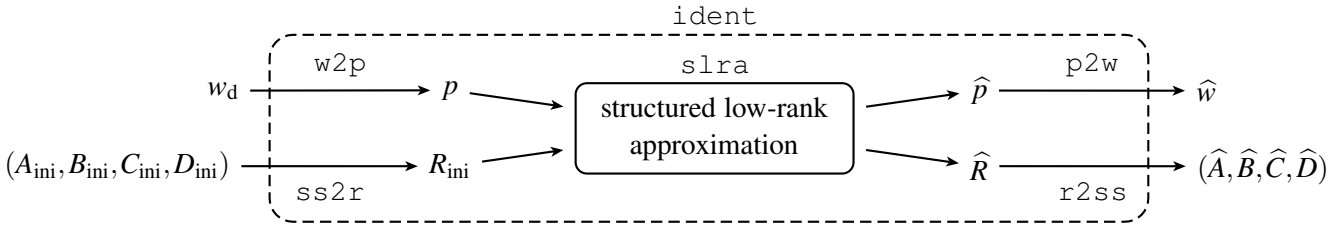

Figure 1: Implementation of the system identification function `ident`.

The function `ident` solves the approximate identification problem (SYSID), and `misfit` computes the misfit $M(w_{\mathrm{d}},\mathscr{B})$. They implement the following mappings (the input/output parameters are defined in Appendix A):

`ident`: $(w_{\mathrm{d}},\mathtt{m},\ell) \mapsto (\widehat{A},\widehat{B},\widehat{C},\widehat{D})$, where $(\widehat{A},\widehat{B},\widehat{C},\widehat{D})$ define a (locally) optimal solution of (SYSID)

⟨`ident` *function definition*⟩≡

```
function [sysh, info, wh, xini] = ident(w, m, ell, opt)
```

`misfit`: $\big(w_{\mathrm{d}},(A,B,C,D)\big) \mapsto (M,\widehat{w})$, where M is the misfit between $\mathscr{B}(A,B,C,D)$ and w_{d}, and $\widehat{w}$ is the optimal approximation of w_{d} within $\mathscr{B}(A,B,C,D)$ (the smoothed trajectory).

⟨`misfit` *function definition*⟩≡

```
function [M, wh, xini] = misfit(w, sysh, opt)
```

4 L_2-optimal model reduction

In this section, we use the `ident` function for solving a (finite horizon) L_2-optimal model reduction problem [5, Section 3.3]. Consider a discrete-time linear time-invariant system $\mathscr{B}$ with an input/output partition $w = (u,y)$ and let $H(t) \in \mathbb{R}^{\mathtt{p}\times\mathtt{m}}$ be the tth sample of the impulse response of $\mathscr{B}$. The finite horizon-T L_2 norm of $\mathscr{B}$ (w.r.t. the given input/output partition $w = (u,y)$) is defined as

$$\|\mathscr{B}\|_{2,T} := \|H\|_{2,T} := \sqrt{\sum_{t=0}^{T} \|H(t)\|_{\mathrm{F}}^2},$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm.

Problem 2 (L_2-optimal model reduction problem [5])**.** Given a linear time-invariant system $\mathscr{B}_{\mathrm{d}} \in \mathscr{L}^q_{\mathtt{m},\ell}$ and a complexity specification $\ell_{\mathrm{red}} < \ell$, find an optimal approximation of $\mathscr{B}_{\mathrm{d}}$ with bounded complexity $(\mathtt{m}, \ell_{\mathrm{red}})$

$$\widehat{\mathscr{B}}^* := \arg\min_{\widehat{\mathscr{B}}} \|\mathscr{B}_{\mathrm{d}} - \widehat{\mathscr{B}}\|_{2,T} \quad \text{subject to} \quad \widehat{\mathscr{B}} \in \mathscr{L}^q_{\mathtt{m},\ell_{\mathrm{red}}}.$$

The link between L_2-optimal model reduction and system identification is due to equivalence of an impulse response of $\mathscr{B}$ and a set of responses of a related autonomous system $\mathscr{B}_{\mathrm{aut}}$.

Proposition 3. *The shifted impulse response*

$$\sigma H = \big(H(1), H(2), \dots\big)$$

of $\mathscr{B} = \mathscr{B}(A,B,C,D)$ is equal to the matrix $\begin{bmatrix} y_1 & \cdots & y_{\mathtt{m}} \end{bmatrix}$ of responses $y_1,\dots,y_{\mathtt{m}}$ of the autonomous system $\mathscr{B}_{aut} = \mathscr{B}(A,C)$ to initial conditions $b_1,\dots,b_{\mathtt{m}}$, where $B = \begin{bmatrix} b_1 & \cdots & b_{\mathtt{m}} \end{bmatrix}$.

Proposition 3 and the fact that $H(0)$ is equal to the feed through parameter D of the input/state/output representation $\mathscr{B}(A,B,C,D)$ imply that the L_2-optimal model reduction problem is equivalent to an L_2-optimal identification problem for an autonomous linear time-invariant system from the set of $\mathtt{m}$ responses $h_1,\dots,h_{\mathtt{m}}$.
A input/state/output representation of the reduced order model is obtained from the identified autonomous system $\mathscr{B}(\widehat{A},\widehat{C})$, initial conditions $X_{\mathrm{ini}} = \begin{bmatrix} \widehat{x}_{\mathrm{ini},1} & \cdots & \widehat{x}_{\mathrm{ini},\mathtt{m}} \end{bmatrix}$, and $H(0)$ as follows:

$$\mathscr{B}\big(\widehat{A}, X_{\mathrm{ini}}, \widehat{C}, H(0)\big).$$

A benchmark model reduction problem

As a test example, consider a mechanical system consisting of N point masses connected in a chain by ideal springs and ideal dampers. The first and the last masses are also connected to walls. Friction force, proportional to the speed, acts on all masses. The parameters of the system are the masses $m_1,\dots,m_N$, the spring and damper coefficients $k_0,k_1,\dots,k_N$ and $d_0,d_1,\dots,d_N$, the length δ of the springs, and the friction coefficient f, which are all nonnegative numbers. Setting the coefficients of the most left and most right springs and dampers to zero has the effect of detaching the chain of masses from the left and right walls.

The system dynamics is described by the system of differential equations

$$\begin{aligned}
m_1\ddot{p}_1 &= -(k_0+k_1)p_1 + k_1p_2 - (d_0+d_1+f)\dot{p}_1 + d_1\dot{p}_2 + (k_0-k_1)\delta,\\
m_2\ddot{p}_2 &= k_1p_1 - (k_1+k_2)p_2 + k_2p_3 + d_1\dot{p}_1 - (d_1+d_2+f)\dot{p}_2 + d_2\dot{p}_3 + (k_1-k_2)\delta,\\
&\vdots\\
m_i\ddot{p}_i &= k_{i-1}p_{i-1} - (k_{i-1}+k_i)p_i + k_ip_{i+1} + d_{i-1}\dot{p}_{i-1} - (d_{i-1}+d_i+f)\dot{p}_i + d_i\dot{p}_{i+1} + (k_{i-1}-k_i)\delta,\\
&\vdots\\
m_N\ddot{p}_N &= k_{N-1}p_{N-1} - (k_{N-1}+k_N)p_N + k_{N-1}\dot{p}_{N-1} - (d_{N-1}+d_N+f)\dot{p}_N + (k_{N-1}+k_NN)\delta.
\end{aligned} \tag{SYS}$$

The positions y of the masses with indexes in the set $\mathscr{Y}$ are observed and external forces u are applied to the masses with indexes in the set $\mathscr{U}$. An input/state/output representation of the resulting model is given in Appendix C.

Numerical results

The input data for the model reduction methods is the first $T+1$ samples of the impulse response H of the full order system and the lag of the reduced order model. The value of T is selected so that, after T steps, the impulse response has converged sufficiently close to zero. The L_2-optimal model reduction problem is solved by the `ident` function and the function `arl2` from the RARL2 package [11].

Note 4 (Using H vs $\mathscr{B}$ as input data). The `arl2` function gives accepts as an input either an input/state/output representation of the model or impulse response coefficients of the system. In the simulations, we use the latter option.

Note 5 (Initial approximation). The optimization based methods `ident` and `arl2` are initialized with the suboptimal approximation computed by Kung's method [3], implemented in the function `fc2ss` of the RARL2 package.

The approximate model $\widehat{\mathscr{B}}$ is evaluated by computing the finite horizon-T relative L_2 approximation error

$$e = \frac{\|\mathscr{B} - \widehat{\mathscr{B}}\|_{2,T}}{\|\mathscr{B}\|_{2,T}}.$$

Example 6. In an example with model parameters

$$\begin{aligned} N = 10, \quad f = 0.1, \quad \delta = 1, \quad \mathscr{U} = 1, \quad \mathscr{Y} = 5, \quad \ell = 3, \\ \begin{bmatrix} m_1 & m_2 & \dots & m_{N-1} & m_N \end{bmatrix} = \begin{bmatrix} 10 & 9 & \cdots & 2 & 1 \end{bmatrix}, \\ \begin{bmatrix} k_0 & k_1 & \dots & k_{N-1} & k_N \end{bmatrix} = \begin{bmatrix} 0.5 & 9 & \cdots & 1 & 0.1 \end{bmatrix}, \\ \begin{bmatrix} d_0 & d_1 & \dots & d_{N-1} & d_N \end{bmatrix} = 0.2 \begin{bmatrix} 0 & 9 & \cdots & 1 & 1 \end{bmatrix}, \end{aligned}$$

the obtained results are:

```
'method'    'kung'      'arl2'      'ident'
'e'         [0.4380]    [0.4263]    [0.4263]
't'         [0.1035]    [2.3251]    [0.0579]
```

The high-order system's impulse responses and the approximations are shown in Figure 2, left. The `arl2` and `ident` functions compute the same approximation, however, the computation time required by the `ident` function is much smaller. This is due to the efficient implementation of the structured low-rank approximation solver.

Example 7. In the setup of Example 6, detaching the system from the right wall ($k_N = d_N = 0$) gives qualitatively similar results:

```
'method'    'kung'      'arl2'      'ident'
'e'         [0.8368]    [0.5950]    [0.5950]
't'         [0.0804]    [3.2495]    [0.0789]
```

The high-order system's impulse response and the approximations are shown in Figure 2, right.

Example 8. In the setup of Example 6, detaching the system from both right and left walls ($k_N = d_N = k_0 = d_0 = 0$) introduced a free motion, *i.e.*, an unstable mode. This gives qualitatively different result in the approximations (see also Figure 3, left):

```
'method'    'kung'      'arl2'      'ident'
'e'         [0.6830]    [0.2244]    [0.0337]
't'         [0.0561]    [1.9533]    [0.0564]
```

The reason for the poor approximation of `arl2` is that this method imposes stability of the approximation.

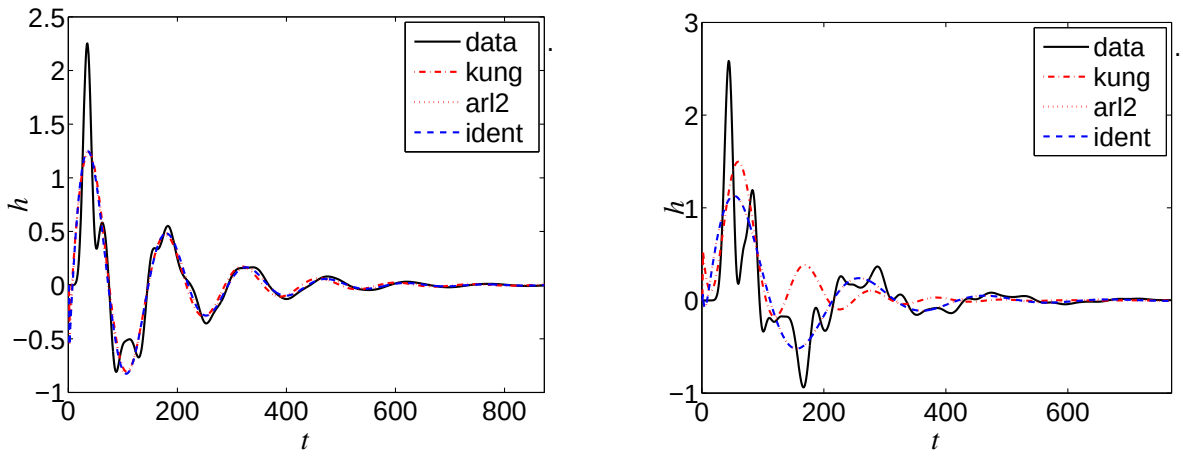

Figure 2: High order and reduced order model responses for model reduction examples 6 and 7.

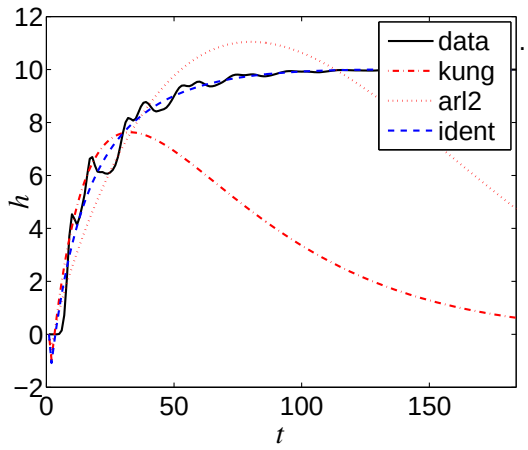

Figure 3: High order and reduced order model responses for model reduction example 8.

5 Identification from multiple trajectories of different lengths

In this section, we test the multiple trajectories feature of the identification method. An alternative method to `ident`, used for comparison, is the function `pem` from the System identification Toolbox of Matlab. The data is generated in the output-error setup and the methods are called with the corresponding options:

- `opt.exct = 1:m` for `ident` and
- `'dist', 'none'` for `pem`.

The true system is randomly generated with `drss` and the true data consists of random trajectories of the true system. The input is normally distributed with zero mean and identity covariance matrix. The data used for identification is a noise corrupted version of the true trajectory, where the output noise is zero mean white Gaussian. The simulation parameters are: number of inputs, specified by the variable `m`; number of outputs, specified by the variable `p`; lag of the system, specified by the variable `ell`; number of time series, specified by the variable `N`; number of samples of the time series, specified by the variable `T`; and a noise standard deviation, specified by the variable `NL`.

The identified models by the `ident` and `pem` functions are compared with respect to their relative output error

$$e = \sum_{i=1}^{N} \frac{\|y^i - \hat{y}^i\|_2^2}{\|y^i\|_2^2}$$

Example 9. In an example with parameters

⟨*Multiple trajectories example 1*⟩≡
```
ex = 'mult-traj-ex1'; ell = 2; p = 1; m = 1; T = [500 1000];
K = 5; NL = linspace(0, 0.1, K); test_mult_traj
```

the results obtained

```
'nl'          [          0]    [0.0250]    [0.0500]    [0.0750]    [0.1000]
'e ident'     [5.5929e-32]    [0.1137]    [0.3360]    [0.5445]    [0.6760]
'e pem'       [3.4563e-32]    [0.1137]    [0.3360]    [0.5445]    [0.6760]
```

show that the `ident` and `pem` methods achieve the same approximation error. (Difference of less than 10^{-4} in the relative approximation errors can be attributed to exiting the optimization with the default convergence tolerance of 10^{-5}.) Although, in this example `ident` and `pem` are functionally equivalent (compute the same answer), `ident` is on the average 28 times faster than `pem`.

Example 10. In the case of $N = 50$ data sets of short ($T = 20$) trajectories

⟨*Multiple trajectories example 2*⟩≡
```
ex = 'mult-traj-ex2'; ell = 2; p = 1; m = 1; T = 20 * ones(1, 50);
K = 5; NL = linspace(0, 0.1, K); test_mult_traj
```

```
'nl'          [          0]    [0.0250]    [0.0500]    [0.0750]    [0.1000]
'e ident'     [2.1716e-32]    [0.0015]    [0.0057]    [0.0130]    [0.0225]
'e pem'       [1.9819e-32]    [0.0015]    [0.0057]    [0.0127]    [0.0225]
```

the obtained results are again identical. In this case the `ident` function is on the average 59 times faster than `pem`. Note that the theoretical computational complexity of the structured low-rank approximation method, used by `ident`, scales linearly in N, see [19]. Empirical results suggest that the algorithms used in the implementation of the `pem` function scale worse with the number of time series.

6 System identification with missing data

In this section, we consider identification problems with missing data. The data w is a noisy T samples long random trajectory of a single-input single-output linear time-invariant system $\bar{\mathscr{B}} = \mathscr{B}(\bar{R})$ with lag $\ell = 2$. Input, output, or both samples are missing at moments of time $t \in \mathscr{T}_{\mathrm{m}}$. The true model parameters are

$$\bar{R}_0 = \begin{bmatrix} 0.3 & 0.7 \end{bmatrix}, \quad \bar{R}_1 = \begin{bmatrix} 1 & -1.4 \end{bmatrix}, \quad \bar{R}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}. \tag{$*$}$$

The approximation accuracy is measured by the angle

$$e = \angle(\bar{R}, \widehat{R}) = \cos^{-1}\left(\frac{\bar{R}^{\top}\widehat{R}}{\|\bar{R}\|\|\widehat{R}\|}\right),$$

between the true $\bar{R}$ and estimated $\widehat{R}$ parameter vectors.

An alternative method for solving the identification problem considered in this section is proposed in [13]. This method uses a frequency domain approach [14]. A Matlab implementation of the algorithm (called below `sysid`) was kindly provided by the authors and is used below for verification of the results obtained with `ident`.

The simulation parameters in the experiments are the number of samples `T`; the set of missing values $\mathscr{T}_{\mathrm{m}}$, specified by a variable `Tm`; and a noise standard deviation interval, specified by a vector `NL`. The reported results show the error e for the compared methods and for the different noise levels specified in `NL`. Three experiments are done for different distribution of the missing values: sequential, periodic, and random. Both input and output values are missing. A `NaN` value in the table of results indicates that the corresponding method fails in (or does not apply to) this case.

Example 11 (Sequential missing data samples). In an example with sequential missing data, the `ident` and `sysid` functions achieve comparable accuracy

⟨*Missing data example 1*⟩≡
```
ex = 'missing-data-ex1';
T = 100; K = 5; NL = linspace(0, 0.01, K); Tm = [30:70]; test_missing_data
```

```
'nl'       [          0]    [0.0025]    [0.0050]    [0.0075]    [0.0100]
'ident'    [2.9802e-08]    [0.0012]    [0.0078]    [0.0017]    [0.0067]
'sysid'    [          0]    [0.0012]    [0.0078]    [0.0014]    [0.0065]
```

The `ident` function is 9 time faster than `sysid`.

Example 12 (Periodic missing data samples). Similar results are obtained in the case of periodic missing data for small noise levels

⟨*Missing data example 2*⟩≡
```
ex = 'missing-data-ex2';
T = 100; K = 5; NL = linspace(0, 0.01, K); Tm = [30:3:70]; test_missing_data
```

```
'nl'       [          0]    [0.0025]    [0.0050]    [0.0075]    [0.0100]
'ident'    [4.0426e-07]    [0.0014]    [0.0058]    [0.0077]    [0.0068]
'sysid'    [          0]    [0.0014]    [0.0073]    [0.0082]    [0.0077]
```

In this example, the `ident` function is 5 time faster than `sysid`.

Example 13 (Randomly distributed missing data samples). Finally, we show a simulation example with $T = 1000$ data points from which 600 are randomly missing

⟨*Missing data example 3*⟩≡
```
ex = 'missing-data-ex3';
T = 1000; K = 5; NL = linspace(0, 0.01, K); Tm = sort(randperm(T, 600));
test_missing_data
```

```
'nl'       [          0]    [0.0025]    [0.0050]    [0.0075]    [0.0100]
'ident'    [9.4278e-05]    [0.0029]    [0.0087]    [0.0028]    [0.0123]
'sysid'    [        NaN]    [   NaN]    [   NaN]    [   NaN]    [   NaN]
```

7 Performance on real-life data

In this section, the performance of the `ident` function is tested on benchmark problems from the data base for system identification DAISY [12]. The data come from a number of applications in process industry, electrical, mechanical, and environmental engineering. We choose a validation criterion that measures the predictive power of the model: how accurate the model can fit a part of the data that is not used for identification. Values for the identification methods' parameters that correspond to this validation criterion are chosen and fixed for all data sets. Although "better" results may be obtained by preprocessing of the data and tuning "hyper parameters" (model structure and identification criterion) this is not done. Our tests reflect the view that the identification process should be as automatic as possible, *i.e.*, it should be done with as little human interaction as possible. We apply the methods on all data sets choosing only the model class (specified by a bound on the model complexity).

The considered data sets are listed in Table 1. References and details about the nature and origin of the data is given in [12]. The data is preprocessed by centering it. The model's lag ℓ is chosen manually for each data set from the complexity–accuracy trade-off curve (misfit as a function of the lag).

The data $w_{\mathrm{d}} = (u_{\mathrm{d}}, y_{\mathrm{d}})$ in all examples is split into identification and validation parts. The first 70% of the data, denoted w_{idt}, are used for identification, and the remaining 30% of the data, denoted w_{val}, are used for validation. A model $\widehat{\mathscr{B}}$ is identified from w_{idt} by an identification method and is validated on w_{val} by the validation criterion defined next. The model class is linear time-invariant systems with a bound ℓ on the lag.

The validation criterion is the "simulation fit" computed by the function `compare` of the System Identification Toolbox. This choice is motivated by the fact that in this case the results obtained by the `ident` function can be validated, using the `pem` function. Correspondingly, the `ident` and `pem` functions are applied setting options for output error identification:

- `opt.exct = 1:m` for `ident` and
- `'dist', 'none'` for `pem`.

#	Data set name	T	m	p	ℓ
1	Data of a simulation of the western basin of Lake Erie	57	5	2	1
2	Data of ethane-ethylene distillation column	90	5	3	1
3	Heating system	801	1	1	2
4	Data from an industrial dryer (Cambridge Control Ltd)	867	3	3	1
5	Data of a laboratory setup acting like a hair dryer	1000	1	1	5
6	Data of the ball-and-beam setup in SISTA	1000	1	1	2
7	Wing flutter data	1024	1	1	5
8	Data from a flexible robot arm	1024	1	1	4
9	Data of a glass furnace (Philips)	1247	3	6	1
10	Heat flow density through a two layer wall	1680	2	1	2
11	Simulation data of a pH neutralization process	2001	2	1	6
12	Data of a CD-player arm	2048	2	2	1
13	Data from a test setup of an industrial winding process	2500	5	2	2
14	Liquid-saturated steam heat exchanger	4000	1	1	2
15	Data from an industrial evaporator	6305	3	3	1
16	Continuous stirred tank reactor	7500	1	2	1
17	Model of a steam generator at Abbott Power Plant	9600	4	4	1

Table 1: Examples from DAISY. T—number of data points, m—number of inputs, p—number of outputs, ℓ—lag of the identified model.

For a given time series $w_{\mathrm{d}} = (u_{\mathrm{d}}, y_{\mathrm{d}})$ and a model $\mathscr{B}$, the approximation $\widehat{y}$ of y in $\mathscr{B}$ is defined as follows:

$$\widehat{y}\big((u_{\mathrm{d}}, y_{\mathrm{d}}), \mathscr{B}\big) := \min_{\widehat{y}} \|y_{\mathrm{d}} - \widehat{y}\| \quad \text{subject to} \quad \operatorname{col}(u_{\mathrm{d}}, \widehat{y}) \in \mathscr{B}.$$

(The optimization is carried over the initial conditions that generate $\widehat{y}$ from the given input u_{d}.) With this notation, the fit of w_{d} by $\mathscr{B}$ is defined as

$$F(w_{\mathrm{d}}, \mathscr{B}) := 100 \, \frac{\max\big(0, 1 - \|y_{\mathrm{d}} - \widehat{y}(w_{\mathrm{d}}, \mathscr{B})\|\big)}{\|y_{\mathrm{d}} - \sum_{t=1}^{T} y_{\mathrm{d}}(t)/T\|}.$$

We compare the fitting criterion $F(w_{\mathrm{val}}, \widehat{\mathscr{B}})$ for the models produced by `ident` and `pem`. The results are shown in Figure 4. In terms of accuracy, the `ident` and `pem` function show comparable performance. The differences in the results can be attributed to the nonconvexity of the optimization problem and the usage of different initial approximations (unstructured low-rank approximation for the `ident` function and a subspace identification method for the `pem` function). In terms of execution time, except for example 17, the `ident` function is faster than `pem`. It should be noted that in example 17, the accuracy achieved by `ident` is significantly higher on both identification and validation data than the one achieved by `pem`.

8 Conclusions

We have presented a method for system identification in the behavioral setting, with the following salient features:

1. representation free problem formulation,
2. identification from multiple time-series,
3. specification of exact variables,
4. missing values in arbitrary variables and moments of time,
5. implementation in a literature programming style.

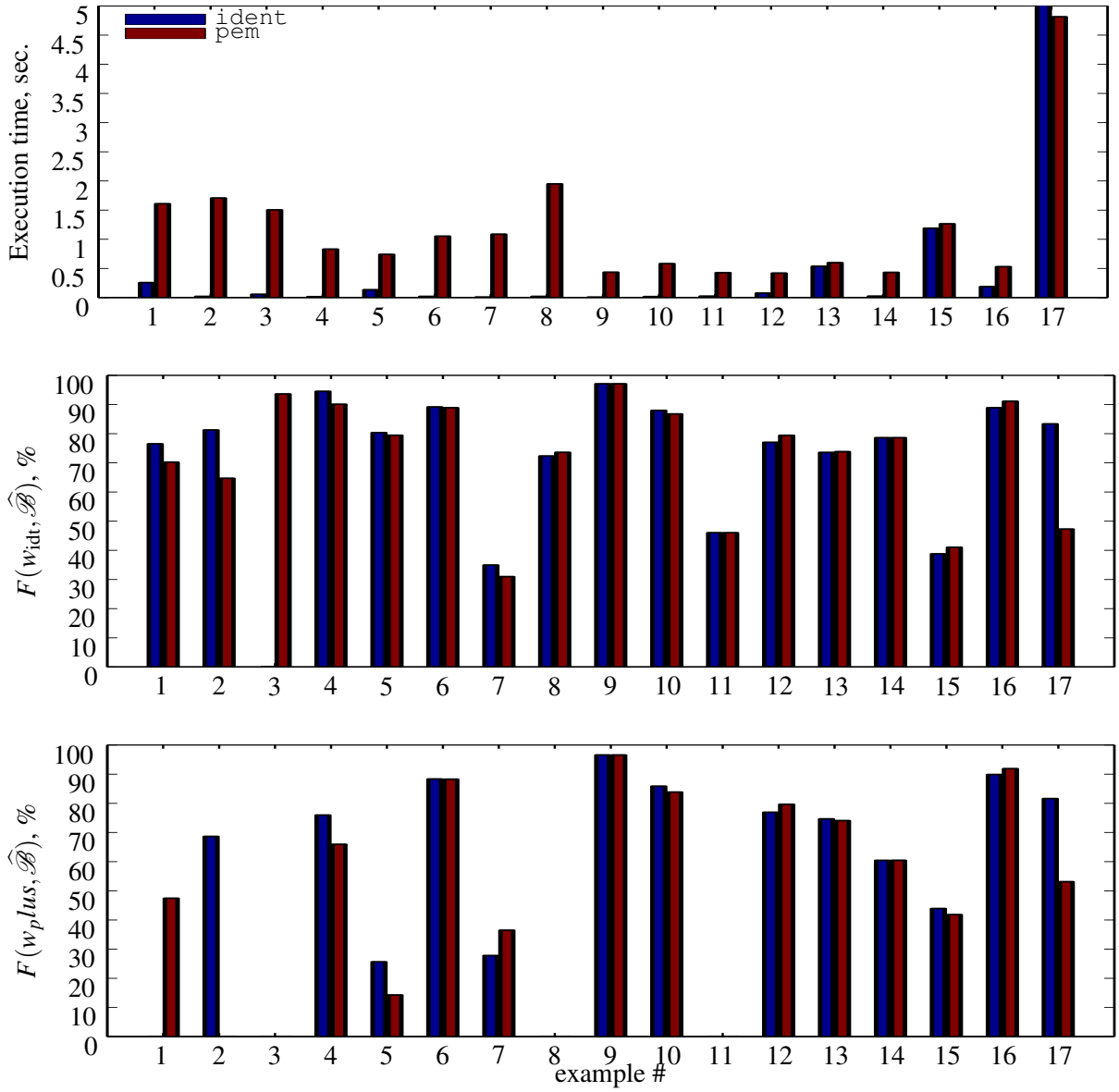

Figure 4: Results on all data sets by splitting of the data into first 70% for identification and remaining 30% for validation.

We showed application of the method for L_2 optimal model reduction, identification from multiple data sets of different length, identification from data with missing values, and benchmark problems from the DAISY dataset. The developed software package was compared with state-of-the-art system identification packages, with respect to accuracy and computational efficiency. Despite the generality and flexibility of the developed software is functionally equivalent to and computationally faster than existing alternatives.

Acknowledgements

The structured low-rank approximation package [7], used in the implementation of the identification method, is a joint work with Konstantin Usevich. During the preparation of the manuscript, I had discussions with Peter Young, Rik Pintelon, Johan Schoukens, Lennart Ljung, and Martine Olivi on various aspects of the results presented in Sections 4–7.

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement number 258581 "Structured low-rank approximation: Theory, algorithms, and applications".

References

[1] M. Galassi et al. *GNU Scientific Library Reference Manual*. 3rd edition.

[2] D. Knuth. *Literate programming*. Cambridge University Press, 1992.

[3] S. Kung. A new identification method and model reduction algorithm via singular value decomposition. In *Proc. 12th Asilomar Conf. Circuits, Systems, Computers*, pages 705–714, Pacific Grove, 1978.

[4] L. Ljung. *System Identification Toolbox: User's guide*. The MathWorks.

[5] I. Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008.

[6] I. Markovsky and P. Rapisarda. Data-driven simulation and control. *Int. J. Control*, 81(12):1946–1959, 2008.

[7] I. Markovsky and K. Usevich. Software for weighted structured low-rank approximation. Technical Report 339974, Univ. of Southampton, `http://eprints.soton.ac.uk/339974`, 2012.

[8] I. Markovsky, S. Van Huffel, and R. Pintelon. Block-Toeplitz/Hankel structured total least squares. *SIAM J. Matrix Anal. Appl.*, 26(4):1083–1099, 2005.

[9] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor. Software for approximate linear system identification. In *Proc. 44th Conf. on Decision and Control*, pages 1559–1564, Seville, Spain, 2005.

[10] I. Markovsky, J. C. Willems, S. Van Huffel, B. De Moor, and R. Pintelon. Application of structured total least squares for system identification and model reduction. *IEEE Trans. Automat. Control*, 50(10):1490–1500, 2005.

[11] Jean-Paul Marmorat and Martine Olivi. RARL2 software (realizations and rational approximation in L_2 norm). `http://www-sop.inria.fr/apics/RARL2`.

[12] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel. DAISY: A database for identification of systems. *Journal A*, 38(3):4–5, 1997. Available from `http://homes.esat.kuleuven.be/~smc/daisy/`.

[13] R. Pintelon and J. Schoukens. Frequency domain system identification with missing data. *IEEE Trans. Automat. Control*, 45(2):364–369, February 2000.

[14] R. Pintelon and J. Schoukens. *System Identification: A Frequency Domain Approach*. IEEE Press, Piscataway, NJ, second edition, 2012.

[15] J. Polderman and J. C. Willems. *Introduction to mathematical systems theory*. Springer-Verlag, New York, 1998.

[16] N. Ramsey. Literate programming simplified. *IEEE Software*, 11:97–105, 1994.

[17] B. Roorda. *Global Total Least Squares—a method for the construction of open approximate models from vector time series*. PhD thesis, Tinbergen Institute, 1995.

[18] J. Schoukens, G. Vandersteen, Y. Rolain, and R. Pintelon. Frequency response function measurements using concatenated subrecords with arbitrary length. *IEEE Transactions on Instrumentation and Measurement*, 61(10):2682–2688, 2012.

[19] K. Usevich and I. Markovsky. Variable projection for affinely structured low-rank approximation in weighted 2-norm, 2012. Available from `http://arxiv.org/abs/1211.3938`.

[20] P. Van Overschee and B. De Moor. *Subspace identification for linear systems: Theory, implementation, applications*. Kluwer, Boston, 1996.

[21] J. C. Willems. From time series to linear system—Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling. *Automatica*, 22, 23:561–580, 675–694, 87–115, 1986, 1987.

[22] J. C. Willems. Paradigms and puzzles in the theory of dynamical systems. 36(3):259–294, 1991.

A Input/output parameters of `ident` and `misfit`

- `w` is the given set of time series w_{d} — a real Matlab array of dimension $T \times q \times N$, where T is the number of samples, q is the number of variables, and N is the number of time series. In case of multiple time series of different dimensions, `w` should be specified as a cell array with N cells, each one of which is a $T_i \times q$ matrix containing the ith time series, *i.e.*,

$$\texttt{w(t, :, k)} = \big(w^k(t)\big)^\top \qquad \text{or} \qquad \texttt{w\{k\}(t,:)} = \big(w^k(t)\big)^\top.$$

- (`m`, `ell`) is the complexity specification (input dimension and lag).
- `opt` is a optional argument specifying exact variables, exact initial conditions, and options for the optimization solver, used by the `ident` function. The options are passed to the functions `ident` and `misfit` as fields of a structure.
  - `'exct'` (default value `[]`) — vector of indices for exact variables.
  - `'wini'` — specifies exact initial conditions. If `wini = 0`, exact zero initial conditions are specified, *i.e.*, $\mathrm{col}(0, \widehat{w}^k) \in \widehat{\mathscr{B}}$, More generally, `wini` $= w_{\mathrm{ini}}$ is an ℓ samples long trajectory (specified by an $\ell \times q \times N$ array or a N-dimensional cell array of $\ell \times q$ matrices), defining initial conditions for the time series $\widehat{w}$, *i.e.*, $\mathrm{col}(w^k_{\mathrm{ini}}, \widehat{w}^k) \in \widehat{\mathscr{B}}$.
  - `'sys0'` — initial approximation: an input/state/output representation of a system, given as an `ss` object, with `m` inputs, `p` $:= q -$ `m` outputs, and order `n` $:= \ell$`p`. Default value is computed by the `slra` function, using unstructured low-rank approximation.
  - Arguments allowing the user to specify different optimization algorithms (`'solver'` and `'method'`), control the displayed information (`'disp'`), and change the convergence criteria are described in the structured low-rank approximation user's manual [7].
- `sysh` is an input/state/output representation of the identified or validated system $\widehat{\mathscr{B}}$, given by an `ss` object.
- `info` is a structure, containing exit information from the structured low-rank approximation solver: `info.M` is the misfit $M(w_{\mathrm{d}}, \widehat{\mathscr{B}})$, `info.time` is the execution time, and `info.iter` is the number of iterations.
- `M` is the misfit $M(w_{\mathrm{d}}, \widehat{\mathscr{B}})$.
- `wh` is the optimal approximating time series.
- `xini` is a matrix whose columns are the initial condition, under which $\widehat{w}^k$, $k = 1, \ldots, N$ are obtained.

B Implementation

B.1 Main functions

Figure 1 visualizes the data processing done when the `ident` function is called. A number of auxiliary functions, explained and defined next, are used. The main one is the structured low-rank approximation solver `slra`, which computes a locally optimal solution of (SLRA). Technically, `slra` is a mex-file calling a C++ solver. For our purposes the `slra` function is a black box with inputs the structure parameter vector p, the structure specification $\mathscr{S}$, upper bound for the rank r, and (optionally) an initial value of the kernel parameter R_{ini}; and outputs the structure parameter vector $\widehat{p}$ of a locally optimal approximation, and the corresponding kernel parameter $\widehat{R}$.

The computation done in the `ident` function is

1. transformation of the user defined data w_{d}, `m`, ℓ, exact variables, and (optionally) initial model into input parameters for the `slra` function; and
2. transformation of the `slra` function's solution $(\widehat{p}, \widehat{R})$ into state-space representation of the optimal approximate model $\widehat{\mathscr{B}} = \mathscr{B}(\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D})$, optimal approximation $\widehat{w}$ of the data w_{d}, and corresponding initial conditions x_{ini}.

Obtaining a input/state/output representation of the identified model is possible in two different ways: 1) using the $\widehat{R}$ parameter and 2) using the trajectory $\widehat{w}$. The first option is a realization problem (see Sections B.2) and the second one is a deterministic identification problem [20]. Both transformations are classic problems, for which solutions exists.

The misfit $M(w_{\mathrm{d}}, \mathscr{B})$ between the data w_{d} and the model $\mathscr{B}$ is the cost function of the approximate identification problem. Its fast computation is a key element of the optimization method used by the `slra` function. A convenient way to access the misfit computation is to call the `ident` function with the data w_{d}, initial approximation corresponding to the model $\mathscr{B}$, and with specification of zero iterations for the optimization solver. Then the misfit is returned in the field `M` of the output parameter `info`.

B.2 $R \mapsto$ input/state/output representation

The transformation from kernel to input/state/output representation is done using the standard observer canonical form, defined in the following proposition.

Proposition 14 ([10, Section IV.A])**.** *Consider a kernel representation $\mathscr{B}(R)$ of a linear time-invariant system and let $R =: \begin{bmatrix} Q & -P \end{bmatrix}$, with*

$$R_i = \begin{bmatrix} Q_i & -P_i \end{bmatrix}, \qquad \text{where} \quad Q_i \in \mathbb{R}^{\mathtt{p}\times\mathtt{m}} \quad \text{and} \quad P_i \in \mathbb{R}^{\mathtt{p}\times\mathtt{p}}.$$

Assuming that P_ℓ is nonsingular, a minimal state-space representation $\mathscr{B}(A,B,C,D)$ of the system $\mathscr{B}(R)$, i.e.*,*

$$\mathscr{B}(A,B,C,D) = \mathscr{B}(\begin{bmatrix} Q & P \end{bmatrix}),$$

is given by

$$A = \begin{bmatrix} 0 & \cdots & 0 & -P_\ell^{-1}P_0 \\ I_{\mathtt{p}} & & & -P_\ell^{-1}P_1 \\ & \ddots & & \\ & & I_{\mathtt{p}} & -P_\ell^{-1}P_{\ell-1} \end{bmatrix}, \quad B = \begin{bmatrix} P_\ell^{-1}(Q_0 - P_0P_\ell^{-1}Q_\ell) \\ P_\ell^{-1}(Q_1 - P_1P_\ell^{-1}Q_\ell) \\ \vdots \\ P_\ell^{-1}(Q_{\ell-1} - P_{\ell-1}P_\ell^{-1}Q_\ell) \end{bmatrix}, \quad C = \begin{bmatrix} 0 & \cdots & 0 & I_{\mathtt{p}} \end{bmatrix}, \quad D = P_\ell^{-1}Q_\ell.$$

B.3 Estimation of the initial state

Given an input/state/output representation $\mathscr{B}(A,B,C,D)$ and a trajectory $w = \operatorname{col}(u,y)$ of that system, the corresponding initial state x_{ini} is computed from the system of linear equations

$$y - y_{\text{forced}} = \mathscr{O}_L(A,C)x_{\text{ini}}, \tag{x_{ini}}$$

where y_{forced} is the output of the system to input u and zero initial conditions and

$$\mathscr{O}_L(A,C) := \operatorname{col}(C, CA, \ldots, CA^{L-1})$$

is the extended observability matrix.

In order to determine uniquely x_{ini} from (x_{ini}), it is sufficient to choose the parameter L equal to the lag ℓ. For $L > \ell$, existence of solution for (x_{ini}) is ca be used as a test for "exactness" of the trajectory $\big(w(1),\ldots,w(L)\big)$ with respect to the model $\mathscr{B}(A,B,C,D)$, *i.e.*, a test for $\big(w(1),\ldots,w(L)\big) \in \mathscr{B}(A,B,C,D)$.

B.4 Input/state/output representation $\mapsto R$

The transformation from input/state/output to kernel representation is done using the following proposition

Proposition 15. *Consider an input/state/output representation $\mathscr{B}(A,B,C,D)$ of a linear time-invariant system with order that is a multiple of the number of outputs. We have,*

$$\mathscr{B}(A,B,C,D) = \mathscr{B}(\begin{bmatrix} Q & -P \end{bmatrix}),$$

where

$$P := \big(\mathscr{O}_{\ell+1}^{\perp}(A,C)\big)^{\top} \qquad \text{and} \qquad Q := P \begin{bmatrix} D & 0 & 0 & \ldots & 0 \\ CB & D & 0 & \ddots & \vdots \\ CAB & CB & D & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ CA^{\ell-2}B & \cdots & CAB & CB & D \end{bmatrix}. \tag{$(A,B,C,D) \mapsto (P,Q)$}$$

(For a matrix M, $M^{\perp}$ is a full column rank matrix of maximal dimension, such that $M^{\perp}M = 0$.)

Proof. The columns of $\mathscr{O}_{\ell+1}(A,C)$ are free trajectories of the system $\mathscr{B}(A,B,C,D)$, so that they are annihilated by the P parameter

$$P\mathscr{O}_{\ell+1}(A,C) = 0.$$

Moreover, by the assumption that the order is a multiple of the number of outputs, we have that

$$\dim\left(\operatorname{leftker}\left(\mathscr{O}_{\ell+1}^{\perp}(A,C)\right)\right) = \mathtt{p}.$$

Therefore, the rows of P form a basis for the left kernel of $\mathscr{O}_{\ell+1}^{\perp}(A,C)$. This proves that

$$P := \left(\mathscr{O}_{\ell+1}^{\perp}(A,C)\right)^{\top}.$$

Consider the sequence of the first $\ell+1$ samples of the impulse response of $\mathscr{B}(A,B,C,D)$, padded with ℓ zeros:

$$Y := \big(\underbrace{0,\ldots,0}_{\ell},D,\underbrace{CB,CAB,\ldots,CA^{\ell-1}B}_{\ell}\big),$$

and the corresponding input sequence

$$U := \big(\underbrace{0,\ldots,0}_{\ell},I_{\mathtt{m}},\underbrace{0,0,\ldots,0}_{\ell}\big).$$

Since (U,Y) is a matrix valued trajectory of the system $\mathscr{B}(\begin{bmatrix}Q & -P\end{bmatrix})$, we have that

$$Q\mathscr{H}_{\ell+1}(U) = P\mathscr{H}_{\ell+1}(Y) \quad \Longrightarrow \quad Q = P\mathscr{H}_{\ell+1}(Y)\begin{bmatrix} & & I_{\mathtt{p}} \\ & \iddots & \\ I_{\mathtt{p}} & & \end{bmatrix}. \tag{$*$}$$

The second equation in $((A,B,C,D)\mapsto(P,Q))$ follows from $(*)$. □

C Input/state/output representation of the model reduction benchmark

Let $p := \operatorname{col}(p_1,\ldots,p_N)$ and define the state vector by

$$x = \operatorname{col}(p,\frac{\mathrm{d}}{\mathrm{d}t}p,x_{N+1}),$$

where x_{N+L} is a constant, $x_{N+1}(t) = x_{N+1}(0)$, for all t. The order of the state space representation is $\mathtt{n} = 2N+1$. Using (SYS), we obtain an input/state/output $\mathscr{B}(A,B,C,D)$ representation with parameters

$$A = \begin{bmatrix} 0 & I_N & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(2N+1)\times(2N+1)}, \quad B = \begin{bmatrix} 0 \\ E_{\mathscr{U}} \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} E_{\mathscr{Y}}^{\top} & 0 & 0 \end{bmatrix}, \quad D = 0, \quad x_{2N+1}(0) = 1,$$

where

$$A_{21} = \begin{bmatrix} -\frac{k_0+k_1}{m_1} & \frac{k_2}{m_1} & & & \\ \frac{k_1}{m_2} & -\frac{k_1+k_2}{m_2} & \frac{k_3}{m_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{k_{N-2}}{m_{N-2}} & -\frac{k_{N-2}+k_{N-1}}{m_{N-2}} & \frac{k_N}{m_{N-2}} \\ & & & \frac{k_{N-1}}{m_N} & -\frac{k_{N-1}+k_N}{m_N} \end{bmatrix}$$

and

$$A_{22} = \begin{bmatrix} -\frac{d_0+d_1+f}{m_1} & \frac{d_2}{m_1} & & & \\ \frac{d_1}{m_2} & -\frac{d_1+d_2+f}{m_2} & \frac{d_3}{m_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{d_{N-2}}{m_{N-2}} & -\frac{d_{N-2}+d_{N-1}+f}{m_{N-2}} & \frac{d_N}{m_{N-2}} \\ & & & \frac{d_{N-1}}{m_N} & -\frac{d_{N-1}+d_N+f}{m_N} \end{bmatrix}, \qquad A_{23} = \begin{bmatrix} \frac{k_0-k_1}{m_1} \\ \frac{k_1-k_2}{m_2} \\ \vdots \\ \frac{k_{N-2}-k_{N-1}}{m_{N-1}} \\ \frac{k_{N-1+Nk_N}}{m_N} \end{bmatrix} \delta.$$

Define the unit vector e_i, as the ith column of the $\mathtt{n} \times \mathtt{n}$ identity matrix. Let $\mathtt{p}$ be the number of elements of the set $\mathscr{Y}$ and $\mathtt{m}$ be the number of elements of the set $\mathscr{U}$. The matrix $E_{\mathscr{U}}$ is the $\mathtt{n} \times \mathtt{m}$ matrix with ith column $e_{\mathscr{U}_i}$ and, similarly, $E_{\mathscr{Y}}$ is the $\mathtt{n} \times \mathtt{p}$ matrix with ith column $e_{\mathscr{Y}_i}$.

The order of the system $\mathscr{B}(A,B,C,D)$ is $2N$, *i.e.*, the state space representation is nonminimal. (It has an uncontrollable mode $x_{N+1} = \text{const.}$)