System identification in the behavioral setting

Ivan Markovsky and Konstantin Usevich

Abstract

Identification problems with no a priori separation of the variables into inputs and outputs and representation invariant approximation criterion are considered. The model class consists of linear time-invariant systems of bounded complexity (number of inputs and order) and the approximation criterion is the minimum of a 2-norm distance between the given time series and a time series that is consistent with the approximate model. The problems are equivalent to and are solved as a mosaic-Hankel structured low-rank approximation problem. Software implementing the approach is developed and compared with other system identification packages on benchmark problems from the DAISY dataset. 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 [Wil87, PW98] 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 [Roo95] and [**?**]. 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 [MWVD04]. 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, state space, convolution kernels, *etc*. A problem can be equivalently posed in frequency domain as well as time domain, using state space, convolution, or any other representation. This fact shows that the choice of the model representation is arbitrary. Different representations have different advantages, so that all of them should be considered as options in solving the problem. The model representation should be viewed as an implementation detail of a particular solution method, rather that a prerequisite for a problem statement. In a representation free system identification problem, the data fitting criterion and the model class are specified in terms of the desired system behavior—set of trajectories of the system.

   An important issue, related to the use of representations in a 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. Different input/output partitions, generically, lead to equivalent system identification problems. 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 advantages of being more general conceptually (the input/output partition is identified together with the model), leading to an easier problem theoretically (solution always exists), and making the computations more robust (ill-conditioning is avoided).

2. *Multiple time-series*

   Another nonstandard feature of the method proposed is the ability to use multiple data sets. Multiple data sets occur in practice when the measurement experiment is repeated or when there are sufficiently many sequential missing data points in one experiment. The problem of using the data available in multiple data sets for identification of a single model is sometimes called "data fusion problem" and has been addressed by various heuristic methods. The special case of time series with equal length has been solved rigorously by a modification of existing methods for a single time series, however, the general case seems to require new methods. We are aware of only one such method [SVRP12], which is based on concatenation of the separate time series into one time series, introducing transient responses at the transitions from one trajectory to another. The method is nonparametric and does not minimize a specified identification criterion. The method presented in the paper is parametric and naturally handles multiple time series.

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 are the data points over ℓ or more 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 variables in the 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. A method for autoregressive exgonous (ARX) models is proposed in [Isa93]. Frequency domain method for single-input single-output systems is presented in [PS00]. 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. In this case, the initial conditions are a priori known to be zeros. 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. When available, a software implementation of the method contains the full details and is therefore the ultimate source of information for the method. 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 [Knu92] comes as a solution to this issue by including the 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 need not be the sequence in which they 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 serve as documentation of the code. We use the `noweb` [Ram94] 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 [Mar08]. Rank deficient mosaic-Hankel matrices [Hei95] 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 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 problem 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 [UM12]. A practical implementation of the methods in C++, using optimization algorithm from the GSL library [G$^+$], is available [MU12]. The identification function presented in this paper is a wrapper function to the structured low-rank approximation solver of [MU12]. Alternative publicly available software packages for system identification and model reduction are:

- System Identification Toolbox of Matlab [Lju],
- CAPTAIN toolbox [TPYT07],
- Frequency domain toolbox, accompanying the book [PS12],
- L_2 model reduction package RARL2 [MO].

In the paper, we show numerical examples comparing the methods in the paper with the alternative ones.

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

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

The number of inputs $\mathtt{m}$, the number of outputs $\mathtt{p}$, 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.

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 $\mathscr{B}$

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

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.*,

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

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

$$w_{\mathrm{d}} = \{w_{\mathrm{d}}^1, \dots, w_{\mathrm{d}}^N\}, \quad \text{where} \quad w_{\mathrm{d}}^k = \big(w_{\mathrm{d}}^k(1), \dots, w_{\mathrm{d}}^k(T_k)\big), \quad \text{for } k = 1, \dots, N,$$

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

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

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}, *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 and a natural number ℓ, we define the mosaic-Hankel matrix (a $q \times N$ block matrix with Hankel blocks)

$$\mathscr{H}_{\ell+1}(w) := \begin{bmatrix} \mathscr{H}_{\ell+1}(w_1^1) & \cdots & \mathscr{H}_{\ell+1}(w_1^N) \\ \vdots & & \vdots \\ \mathscr{H}_{\ell+1}(w_q^1) & \cdots & \mathscr{H}_{\ell+1}(w_q^N) \end{bmatrix}, \quad \text{where} \quad \mathscr{H}_{\ell+1}(w_i^k) := \begin{bmatrix} w_i^k(1) & w_i^k(2) & \cdots & w_i^k(T-\ell) \\ w_i^k(2) & w_i^k(3) & \cdots & w_i^k(T-\ell+1) \\ \vdots & \vdots & & \vdots \\ w_i^k(\ell+1) & w_i^k(\ell+2) & \cdots & w_i^k(T) \end{bmatrix}.$$

The identification problem (SYSID) is solved in the following equivalent formulation:

$$\widehat{R} = \arg \min_{R, RR^\top = I_{\mathtt{p}}} \left(\min_{\widehat{w}} \|w_{\mathrm{d}} - \widehat{w}\|_{\ell_2}^2 \quad \text{subject to} \quad R\mathscr{H}_{\ell+1}(\widehat{w}) = 0 \right). \qquad \text{(SLRA)}$$

The parameter R of (SLRA) is related to the parameters $R_0, R_1, \dots, R_\ell$ of the difference equation representation (DE) of the approximating system $\mathscr{B}$ 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 [MVP05, Mar12]. We use the C++ solver of [MU12] as the core computational tool for solving the system identification problem (SYSID). 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. On its own, the 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 $\mathscr{B}_{\mathrm{ini}} \in \mathscr{L}_{\mathtt{m},\ell}^q$, 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. 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.

Usage

The function `ident` solves the approximate identification problem (SYSID), and `misfit` computes the misfit $M(w_{\rm d},\mathscr{B})$. Both functions use the input/state/output representation (I/S/O), so that they implement the following mappings:

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

5a ⟨`ident` *function definition* 5a⟩≡ (? 0—1)
```
function [sysh, info, wh, xini] = ident(w, m, ell, varargin)
```

`misfit`: $\big(w_{\rm d},(A,B,C,D)\big) \mapsto (M,\widehat{w})$, where M is the misfit between the model, defined by (A,B,C,D), and $w_{\rm d}$.

5b ⟨`misfit` *function definition* 5b⟩≡ (? 0—1)
```
function [M, wh, xini] = misfit(w, sysh, varargin)
```

- `w` is the given time series $w_{\rm d}$ — 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.*,
$$\mathtt{w(t,\,:,\,k)} = \big(w^k(t)\big)^\top \qquad \text{or} \qquad \mathtt{w\{k\}(t,:)} = \big(w^k(t)\big)^\top.$$
- (`m`, `ell`) is the complexity specification (input dimension and lag).
- `varargin` is/are optional arguments(s) specifying exact variables, exact initial conditions, and options for the optimization solver, used by the `ident` function. The options can be passed to the functions `ident` and `misfit` as fields of a variable or as pairs (`'name', 'value'`) of input arguments.
  - `'exct'` (default value `[]`) — vector of indices for exact variables.
  - `'wini'` (default value `[]`) — specifies exact initial conditions. If `wini = []`, initial conditions are not specified. If `wini = 0`, exact zero initial conditions are specified, *i.e.*, $\left[\begin{smallmatrix}0\\ \widehat{w}^k\end{smallmatrix}\right] \in \widehat{\mathscr{B}}$, More generally, $\mathtt{wini} = w_{\rm ini}$ 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.*, $\left[\begin{smallmatrix}w^k_{\rm ini}\\ \widehat{w}^k\end{smallmatrix}\right] \in \widehat{\mathscr{B}}$.
  - `'sys0'` — initial approximation: an input/state/output representation of a system, given as a state space (`ss`) object, with `m` inputs, $\mathtt{p} := q - \mathtt{m}$ outputs, and order $\mathtt{n} := \ell\mathtt{p}$. (Default value is computed by unstructured low-rank approximation.)
  - Arguments allowing the user to specify different optimization algorithms (`'method'`), control the displayed information (`'disp'`), and change the convergence criteria (`'tolX'`, `'tolGrad'`, and `'maxiter'`) are described in the structured low-rank approximation user's manual [MU12].
- `sysh` is an input/state/output representation of the identified or validated system $\widehat{\mathscr{B}}$.
- `info` is a structure, containing exit information from the structured low-rank approximation solver: `info.M` is the misfit $M(w_{\rm d},\widehat{\mathscr{B}})$, `info.time` is the execution time, and `info.iter` is the number of iterations.
- `M` is the misfit $M(w_{\rm d},\widehat{\mathscr{B}})$.
- `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.

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 [Mar08, 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

$$\|\mathscr{B}\|_{2,T} := \|H\|_{2,T} := \sqrt{\sum_{t=0}^{T} \|H(t)\|_2^2}.$$

Problem 1 (L_2-optimal model reduction problem [Mar08])**.** 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 2. *The shifted impulse response*

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

of $\mathscr{B} = \mathscr{B}_\mathrm{ss}(A,B,C,D)$ *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* $\mathscr{B}_{aut} = \mathscr{B}_\mathrm{ss}(A,C)$ *to initial conditions* $b_1,\dots,b_\mathtt{m}$*, where* $B = \begin{bmatrix} b_1 & \cdots & b_\mathtt{m}\end{bmatrix}$.

Proposition 2 and the fact that $H(0)$ is equal to the feed through parameter D of the state space representation $\mathscr{B}_\mathrm{ss}(A,B,C,D)$ 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 m responses $h_1,\dots,h_\mathtt{m}$.

6a ⟨*model reduction by* `ident` 6a⟩≡ (? 0—1)
```
[sysh, info, yh, xini] = ident(y(2:end), 0, ell, opt);
```
⟨$(\mathscr{B}_\mathrm{ss}(\widehat{A},\widehat{C}),\widehat{X}_\mathrm{ini}) \mapsto \mathscr{B}_\mathrm{ss}(\widehat{A},\widehat{B},\widehat{C},\widehat{D})$ 6b⟩

A state space representation of the reduced order model is obtained from the identified autonomous system $\mathscr{B}_\mathrm{ss}(\widehat{A},\widehat{C})$, 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:

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

6b ⟨$(\mathscr{B}_\mathrm{ss}(\widehat{A},\widehat{C}),\widehat{X}_\mathrm{ini}) \mapsto \mathscr{B}_\mathrm{ss}(\widehat{A},\widehat{B},\widehat{C},\widehat{D})$ 6b⟩≡ (6a)
```
sysh.b = xini; sysh.d = y(1);
```

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/or most right springs and dampers to zero has the effect of detaching the chain of masses from the left and/or right walls.

The system dynamics is described by the system of differential equations

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

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 $\mathscr{U}$.

The dynamics of the system can be represented by an input/state/output representation (I/S/O) with parameters

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

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

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

The model is multivariable dynamical system of order $\mathtt{n} = 2N$.

Simulation examples

The input data for the model reduction methods is the first $T+1$ samples of the impulse response H of the system $\mathscr{B}_{\text{ss}}(A,B,C,D)$, with parameters defined in (1), and the lag of the reduced 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 [MO].

7a ⟨*model reduction by* `arl2` 7a⟩≡ (? 0—1)
```
addpath ~/mfiles/rarl2/arl2lib ~/mfiles/rarl2/boplib
arl2_options = arl2Options('real');
optim_options = optimset( ...
    'MaxFunEval', 1e4, 'MaxIter', 1e3, ...
    'TolFun', 1e-12, 'TolX', 1e-12, ...
    'GradObj', 'on', 'DerivativeCheck', 'off', 'Display', 'off');
report = arl2(struct('type', 'coeff', ...
                     'value', reshape(y(2:end), 1, 1, T), ...
                     'valinf', y(1)), ...
                      opt.sys0, arl2_options, optim_options);
sysh = report.sys;
```

The package gives an option for the user to supply as input data either a state space representation of the model or impulse response coefficients of the system. In the simulations, we use the second option. The optimization based methods `ident` and `arl2` are initialized with the suboptimal approximation computed by Kung's method [Kun78] (implemented in the RARL2 package).

7b ⟨*model reduction by* `fc2ss` 7b⟩≡ (? 0—1)
```
addpath ~/mfiles/rarl2/arl2lib; sysh = fc2ss(y(2:end), y(1), ell);
```

The approximate models are compared by computing the finite horizon-T relative L_2 approximation error

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

7c ⟨*evaluate* 7c⟩≡ (? 0—1)
```
yh = impulse(sysh, T); plot(yh, ls), e = norm(y - yh) / norm(y);
```

For an example with model parameters

8 ⟨*Model reduction example 1* 8⟩≡
```
ex = 'ex-mod-red1'; N = 15; opt.f = 0.1; ell = 3;
opt.M = (N:-1:1)';
opt.K = [.5; ones(N - 1, 1); .1];
opt.D = 0.2 * [0; ones(N - 1, 1); 1]; test_mod_red
```
the obtained results are:

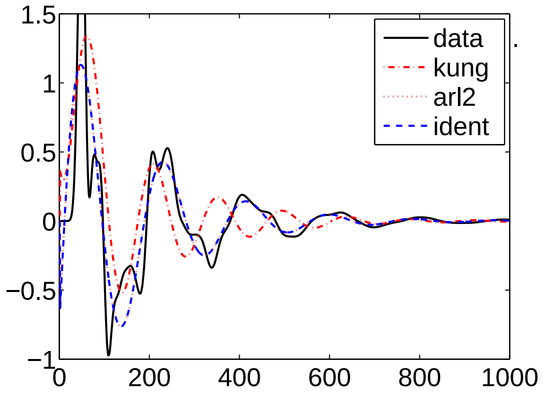
```
'method'    'kung'      'arl2'      'ident'
'e'         [0.7805]    [0.5663]    [0.5663]
't'         [0.0820]    [3.5695]    [0.0621]
```

The high-order systems impulse responses and the approximations are shown in Figure 1, 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.

In the same setup, detaching the system from the right wall gives qualitatively similar results:

```
'method'    'kung'      'arl2'      'ident'
'e'         [0.8368]    [0.5950]    [0.5950]
't'         [0.0715]    [2.7752]    [0.3482]
```

The high-order systems impulse responses and the approximations are shown in Figure 1, right.



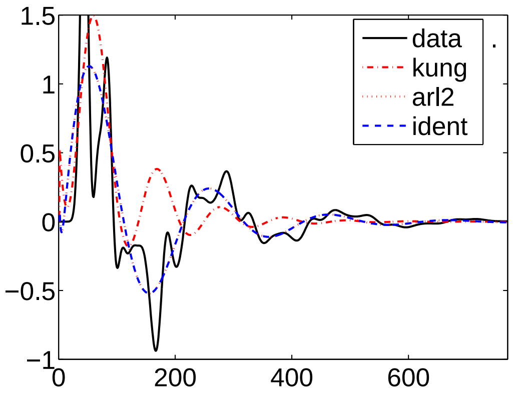
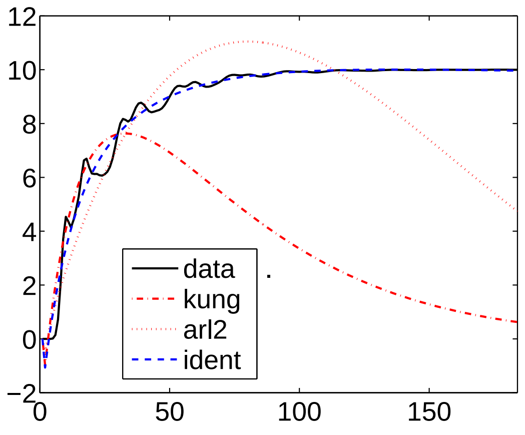


Figure 1: Results for model reduction examples 1 and 2.

Finally, detaching the system from both the right as well as the left walls introduced an a free motion, *i.e.*, an unstable mode. This gives qualitatively different result in the approximations (see Figure **??**):

```
'method'    'kung'      'arl2'      'ident'
'e'         [0.6830]    [0.2244]    [0.0337]
't'         [0.0604]    [1.9900]    [0.0998]
```

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



5 Identification from multiple trajectories of different lengths

In this section, we test the multiple trajectories feature of the identification method.

9a ⟨*use* ident 9a⟩≡ (? 9f)

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

An alternative method used for comparison is the `pem` function from the System identification Toolbox of Matlab.

9b ⟨*use ID toolbox* 9b⟩≡ (9f)

```
data = iddata(w{1}(:, 2), w{1}(:, 1));
for k = 2:N, data = merge(data, iddata(w{k}(:, 2), w{k}(:, 1))); end
sysh = pem(data, ell, 'dist', 'none');
```

The simulation parameters are:

9c ⟨*Multiple trajectories* 9c⟩≡ 9d▷

```
ell = 2; p = 1; m = 1; T = [10 10 10]; nl = 0.0; eiv = 0; opt = struct;
```

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 a zero mean and identity covariance matrix and depending on the value of `opt.ini`, the initial conditions are normally distributed with zero mean and identity covariance matrix or zeros.)

9d ⟨*Multiple trajectories* 9c⟩+≡ ◁9c 9e▷

```
sys0 = drss(ell * p, p, m); N = length(T);
if isfield(opt, 'wini') && opt.wini ~= 0;
  xini = randn(p * ell, 1);
else
  xini = zeros(p * ell, 1);
end
for k = 1:N,
  u0{k} = randn(T(k), 1); % [eye(m); zeros(T(k) - m, m)]; % randn, ones
  y0{k} = lsim(sys0, u0{k}, [], xini); w0{k} = [u0{k} y0{k}];
end
```

The data used for identification is a noise corrupted version of the true trajectory.

9e ⟨*Multiple trajectories* 9c⟩+≡ ◁9d 9f▷

```
if eiv, opt.exct = []; else opt.exct = 1:m; end
v = ones(m + p, 1); v(opt.exct) = 0;
for k = 1:N
  wn = randn(size(w0{k})) * diag(v);
  w{k} = w0{k} + (nl * norm(w0{k}) / norm(wn)) * wn;
end
if N == 1, w0 = w0{1}; w = w{1}; end
```

The identified models by the `ident` and `pem` functions are compared with respect to their relative distances to the data *w*.

9f ⟨*Multiple trajectories* 9c⟩+≡ ◁9e

```
tic, ⟨use ID toolbox 9b⟩, t_pem  = toc; ⟨test multiple trajectories 9g⟩, e_pem = e;
tic, ⟨use ident 9a⟩, t_ident = toc; ⟨test multiple trajectories 9g⟩, e_ident = e;
e = 0; for k = 1:N, e = e + norm(w{k}(:, 2)) ^ 2; end
res = [{'method'; 'e'; 't'} [{'pem' 'ident'};
              num2cell([e_pem e_ident] / e)
              num2cell([t_pem t_ident])]];
diary('ex-mult-traj'), disp(res), diary off
```

9g ⟨*test multiple trajectories* 9g⟩≡ (9f)

```
yh  = compare(data, sysh);
e = 0; for k = 1:N, e = e + norm(w{k}(:, 2) - yh{k}.y) ^ 2; end
```

The obtained results are:

```
'method'    'pem'       'ident'
'e'         [0.3697]    [1.1943e-04]
't'         [3.0303]    [    0.1076]
```

6 System identification with missing data

The data w is a noisy $T = 100$ samples long random trajectory of a single-input single-output linear time-invariant system $\bar{\mathscr{B}} = \mathscr{B}(\bar{R})$ with lag $\ell = 2$. Samples $w(t)$, for $t \in \mathscr{T}_{\mathrm{m}}$, are missing. The true model parameters are

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

The approximation accuracy is measured by the relative L_2 norm of the error system:

$$e = \frac{\|\bar{\mathscr{B}} - \widehat{\mathscr{B}}\|_2}{\|\bar{\mathscr{B}}\|_2}.$$

10a ⟨*evaluate SYSID* 10a⟩≡ (? 0—1)
```
e = norm(sys0 - sysh) / norm(sys0);
```

The identification problem is solved by the following methods:

- Structured low-rank approximation
- System identification toolbox

10b ⟨*use* `misdata` 10b⟩≡ (? 0—1)
```
idw = misdata(iddata(w(:, 2), w(:, 1)), idss(ell));
sysh = tf(oe(idw, [2 2 1])); wh = [w(:, 1) lsim(sysh, w(:, 1))];
```

- Method of [PS00]

10c ⟨`sysid` 10c⟩≡
```
addpath ~/mfiles/missing-data-dynamic/rik/
try
  [sysh, wh_m] = sysid_missing_data_rik(w(:, 1), w(:, 2), ell, [0 std(yt)]);
  wh = zeros(size(w)); wh(Tm, m_io) = wh_m;
catch,
  wh_m = NaN; Rh = NaN;
end
```

- CAPTAIN toolbox

10d ⟨*use* `captain` 10d⟩≡ (? 0—1)
```
addpath ~/mfiles/captain/
[th, stats, e, var, Ps, Pc, yh] = riv([w(:, 2), w(:, 1)], [ell ell 1 0]);
[Ph, Qh] = getpar(th); sysh = tf(Qh, Ph, -1); wh_m = yh(Tm, 1);
```

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 noise variance interval, specified by a vector `NL`. The reported results show the estimation error e for the compared methods and for the different noise levels specified in `NL`. In the case of uniformly distributed missing data samples:

10e ⟨*SYSID example 1* 10e⟩≡
```
ex = 'ex-sysid'; T = 100; N = 7; NL = linspace(0, 0.05, N); Tm = 30:3:70; m_io = 2; test_sy
```

```
'nl'       [         0]    [0.0200]    [0.0400]    [0.0600]    [0.0800]    [0.1000]
'ident'    [2.1880e-06]    [0.0116]    [0.0123]    [0.0299]    [0.0269]    [0.0517]
'sysid'    [    0.2757]    [0.2741]    [0.2619]    [0.2602]    [0.2799]    [0.2591]
'cap'      [    0.0248]    [0.0203]    [0.0280]    [0.0221]    [0.0600]    [0.0469]
```

7 Performance on real-life data

In this section, the performance of the method, described in the paper, is tested on benchmark problems from the data base for system identification DAISY [MGSF97]. The data come from a number of applications: process industry, electrical, mechanical, and environmental. 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 often better results can be obtained by (preprocessing of the data and tuning of the identification method parameters), our tests reflect the view that the identification process should be done with as little human interaction as possible. We apply the methods choosing only the model class (specified by a bound on the model complexity) and the identification/validation criterion (specified by desired notion of approximation or disturbance/noise properties).

7.1 Database for system identification DAISY

The considered data sets are listed in Table 1. References and details about the nature and origin of the data is given in Appendix **??**. The data is preprocessed only 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).

#	Data set name	T	$\mathtt{m}$	$\mathtt{p}$	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, $\mathtt{m}$—number of inputs, $\mathtt{p}$—number of outputs, ℓ—lag of the identified model.

The data $w_{\text{d}} = (u_{\text{d}}, y_{\text{d}})$ in all examples is split into identification and validation parts. For a chosen $x \in [0, 100]$, the first or last x% of the data, denoted w_{idt}, are used for identification, and the remaining $(100 - x)$% 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 (degree of a difference equation representation or equivalently observability index). Bounding the lag by ℓ corresponds to bounding the order by $\ell\mathtt{p}$, where $\mathtt{p}$ is the number of outputs.

7.2 Validation criterion and results

The validation criterion is the "simulation fit" computed by the function `compare` of the System Identification Toolbox. Given a time series $w_{\text{d}} = (u_{\text{d}}, y_{\text{d}})$ and a model $\mathscr{B}$, define the approximation $\widehat{y}$ of y in $\mathscr{B}$ as follows:

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

(The optimization is carried over the initial conditions that generate $\widehat{y}$ from the given input u_{d}.) Let $\bar{y}$ be the mean of y_{d}, *i.e.*, $\bar{y} := \sum_{t=1}^{T} y_{\mathrm{d}}(t)/T$. With this notation, the fit of w_{d} by $\mathscr{B}$ is defined as

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

We compare the fitting criterion $F(w_{\mathrm{val}}, \widehat{\mathscr{B}})$ for the models produced by the compared identification methods.

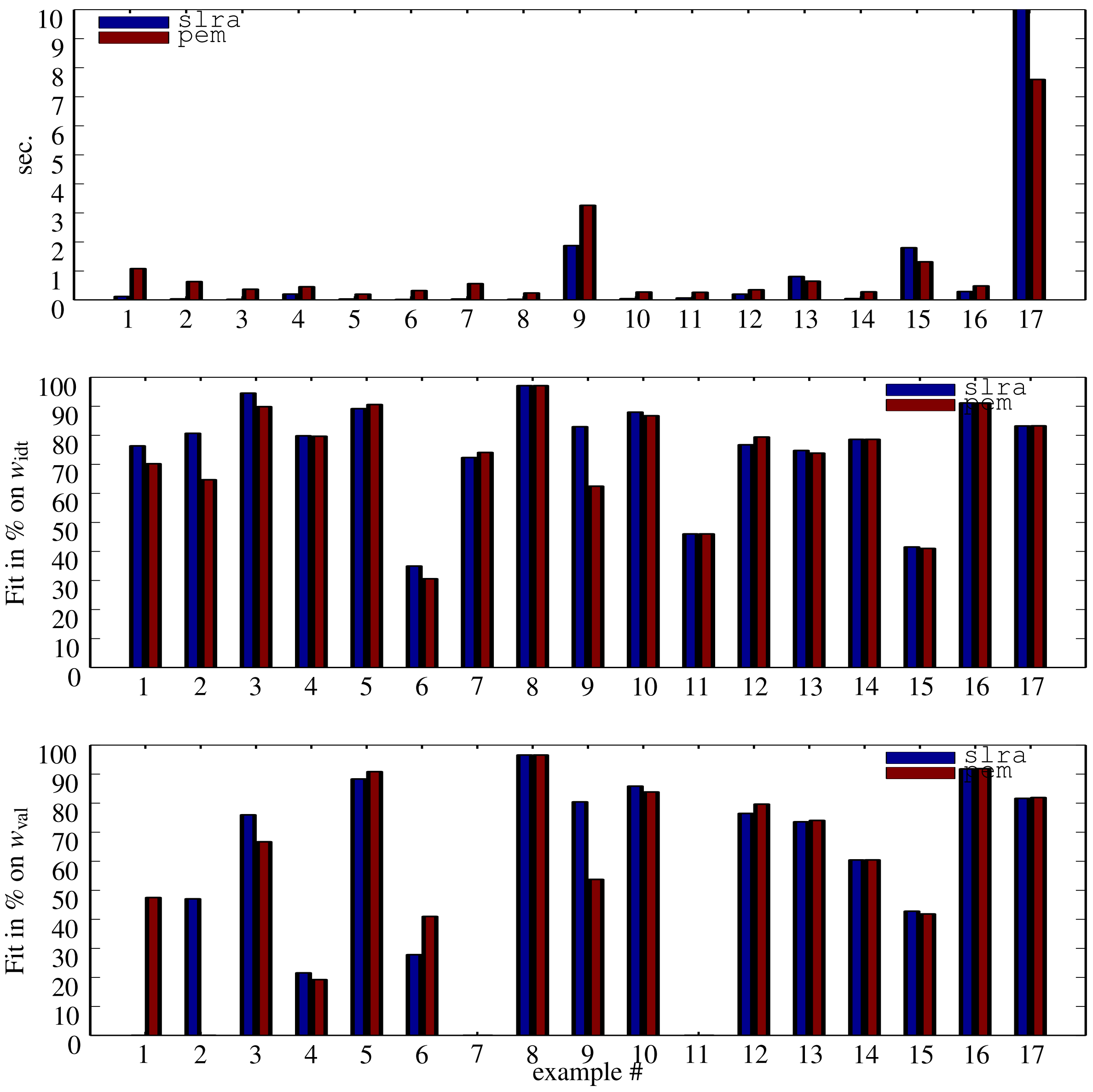


Figure 2: Results on all data sets by splitting of the data into first 70% for identification and remaining 30% for validation.

Acknowledgements

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

[G$^+$] M. Galassi et al. *GNU Scientific Library Reference Manual.* 3rd edition.

[Hei95] G. Heinig. Generalized inverses of Hankel and Toeplitz mosaic matrices. *Linear Algebra Appl.*, 216(0):43–59, February 1995.

[Isa93] A. Isaksson. Identification of ARX-models subject to missing data. *IEEE Trans. Automat. Control*, 38(5):813–819, May 1993.

[Knu92] D. Knuth. *Literate programming.* Cambridge University Press, 1992.

[Kun78] 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.

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

[Mar08] I. Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008.

[Mar12] I. Markovsky. *Low Rank Approximation: Algorithms, Implementation, Applications.* Springer, 2012.

[MGSF97] 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/`.

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

[MU12] 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.

[MVP05] 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.

[MWVD04] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor. Software for approximate linear system identification. Technical Report 04–221, Dept. EE, K.U.Leuven, 2004.

[MWVD05] 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.

[PS00] R. Pintelon and J. Schoukens. Frequency domain system identification with missing data. *IEEE Trans. Automat. Control*, 45(2):364–369, February 2000.

[PS12] R. Pintelon and J. Schoukens. *System Identification: A Frequency Domain Approach.* IEEE Press, Piscataway, NJ, second edition, 2012.

[PW98] J. Polderman and J. C. Willems. *Introduction to mathematical systems theory.* Springer-Verlag, New York, 1998.

[Ram94] N. Ramsey. Literate programming simplified. *IEEE Software*, 11:97–105, 1994.

[Roo95] B. Roorda. *Global Total Least Squares—a method for the construction of open approximate models from vector time series.* PhD thesis, Tinbergen Institute, 1995.

[SVRP12] 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.

[TPYT07] C. J. Taylor, D. J. Pedregal, P. C. Young, and W. Tych. Environmental time series analysis and forecasting with the CAPTAIN toolbox. *Environmental Modelling & Software*, 22:797âĂŞ–814, 2007.

[UM12] 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`.

[Wil87] 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.