Common dynamics estimation

Ivan Markovsky, Tianxiang Liu, and Akiko Takeda

Abstract

Estimation of common dynamics among several observed signals occurs in signal processing, system theory, and computer algebra problems. In this paper, we propose optimization and subspace methods for common linear time-invariant dynamics detection and estimation. First, we consider the deterministic problem of detection of common dynamics when the data is exact (noise free). Then, we consider the stochastic estimation problem when the data is corrupted by white Gaussian noise. The methods proposed have a system theoretic interpretation of finding the intersection of autonomous linear time-invariant behaviors. The subspace identification methods developed are computationally fast but statistically less accurate than alternative optimization methods. Development of local optimization-based methods for common dynamics estimation is a topic of future work.

Index Terms

Common dynamics, Subspace identification, Behavioral approach, Approximate common divisor.

I. Introduction

A prototypical signal processing problem is the one of separating signal from additive measurement noise. Without prior knowledge about the signal and the noise, the separation problem is ill-posed, *i.e.*, it has a nonunique solution. First, we outline the classical setup and corresponding assumptions that make the signal from noise separation problem well-posed. The signal is assumed to be a trajectory of a low-complexity linear time-invariant system and the noise is assumed to be zero-mean white Gaussian process. A technical condition that allows separation is persistency of excitation of the true signal. Then, we describe a generalization of the classical setup where the signal is vector valued and the noise has two components: a trajectory of a low-complexity linear time-invariant system (*structured noise*) and zero-mean white Gaussian process (*unstructured noise*). The key assumption that allows separability in this case is that the structured noise model has no common poles. In addition, as in the classical setup, a persistency of excitation condition is needed. The estimation problem in the new setup leads to common dynamics estimation.

I. Markovsky is with the Department ELEC of the Vrije Universiteit Brussel, Brussels 1050, Belgium (e-mail: ivan.markovsky@vub.ac.be), T. Liu is with RIKEN Center for Advanced Intelligence Project, Nihonbashi, Chuo-ku, Tokyo 103–0027, Japan (e-mail: tianxiang.liu@riken.jp), and A. Takeda is with the Department of Creative Informatics, The University of Tokyo, Tokyo, 113-8656, Japan (e-mail: takeda@mist.i.u-tokyo.ac.jp).

A. Classical setup: zero mean white Gaussian measurement noise

Classical assumptions that make the signal from noise separation problem well-posed are that the true signal is generated by a low-complexity linear time-invariant system and the noise is zero-mean white Gaussian. Let y_{d} be the observed signal, $\bar{y}$ its true value, and e the measurement noise. Using the behavioral language [1], [2], [3], we write concisely the fact that $\bar{y}$ is generated by a linear time-invariant system of order at most $\bar{n}$ as $\bar{y} \in \bar{\mathscr{B}} \in \mathscr{L}_{\bar{n}}$. Here $\bar{\mathscr{B}}$ is the true model and $\mathscr{L}_{\bar{n}}$ is the class of linear time-invariant systems of order at most $\bar{n}$. Let $e \sim \mathrm{N}(0, \sigma^2 I)$ denote a zero mean Gaussian process with covariance matrix $\sigma^2 I$. With this notation, the classical data generating model is

$$y_{\mathrm{d}} = \bar{y} + e, \quad \text{where} \quad \bar{y} \in \bar{\mathscr{B}} \in \mathscr{L}_{\bar{n}} \text{ and } e \sim \mathrm{N}(0, \sigma^2 I). \tag{1}$$

The maximum likelihood estimation problem for (1) is: Given y_{d} and $\bar{n}$, find estimates $\widehat{y}$ of $\bar{y}$ and $\widehat{\mathscr{B}}$ of $\bar{\mathscr{B}}$ as a solution of the optimization problem

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{y} \text{ and } \widehat{\mathscr{B}} \quad \|y_{\mathrm{d}} - \widehat{y}\| \\ &\text{subject to} \quad \widehat{y} \in \widehat{\mathscr{B}} \in \mathscr{L}_{\bar{n}}, \end{aligned} \tag{2}$$

where $\|\cdot\|$ is the 2-norm. In applications of simulation, prediction, and control, of interest is the estimate $\widehat{\mathscr{B}}$ of the model rather than the estimate $\widehat{y}$ of the signal $\bar{y}$.

In the case of noise free data, *i.e.*, $y_{\mathrm{d}} = \bar{y}$, the signal from noise separation problem has a trivial solution $\widehat{y} = y_{\mathrm{d}}$. Nevertheless, the data modeling problem $y_{\mathrm{d}} \mapsto \widehat{\mathscr{B}}$ is meaningful and nontrivial. The main question is: Under what conditions on the data and the data generating model, the true system $\bar{\mathscr{B}}$ can be recovered back from the data y_{d}, *i.e.*, under what conditions $\widehat{\mathscr{B}} = \bar{\mathscr{B}}$? For (1), a necessary and sufficient identifiability condition is that the true signal $\bar{y}$ is persistently exciting of order $\bar{n}$, *i.e.*, $\bar{y}$ can not be fitted exactly by a system of order less than $\bar{n}$.

A linear time-invariant system admits different representations. In this paper, we use the state-space, polynomial, also called kernel, and poles, also called sum-of-damped-exponentials, representations. When the model $\widehat{\mathscr{B}}$ is represented in the state space, the exact modeling problem is equivalent to the system realization problem (*i.e.*, the problem of obtaining a state space representation of the system from its impulse response).

The subproblem of (2), where the model $\widehat{\mathscr{B}}$ is given,

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{y} \quad \|y_{\mathrm{d}} - \widehat{y}\| \\ &\text{subject to} \quad \widehat{y} \in \widehat{\mathscr{B}} \end{aligned} \tag{3}$$

is called the *smoothing problem*. An efficient way to solve it is the Kalman filter, which effectively employs a state space representation of the model $\widehat{\mathscr{B}}$. Alternatively, the Kalman filter can be viewed as an estimation method for the initial conditions. In case of a noise free data signal $y_{\mathrm{d}} \in \widehat{\mathscr{B}}$, (3) has the trivial solution $\widehat{y} = y_{\mathrm{d}}$. However, the initial state estimation problem is nontrivial and meaningful; it is the observer design problem.

The assumption $\bar{y} \in \bar{\mathscr{B}} \in \mathscr{L}_{\bar{n}}$ that the true signal is generated by a low-complexity linear time-invariant system can be expressed equivalently as a constraint on the rank of a Hankel matrix constructed from the signal

$$\operatorname{rank} \underbrace{\begin{bmatrix} \bar{y}(1) & \bar{y}(2) & \cdots & \bar{y}(T-\bar{n}) \\ \bar{y}(2) & \bar{y}(3) & \cdots & \\ \vdots & \vdots & & \vdots \\ \bar{y}(\bar{n}+1) & \bar{y}(\bar{n}+2) & \cdots & \bar{y}(T) \end{bmatrix}}_{\mathscr{H}_{\bar{n}+1}(\bar{y})} \leq \bar{n}.$$

When the rank is maximal, the signal is persistently exciting. The maximum-likelihood estimation problem (2) is equivalent to the following Hankel structured low-rank approximation problem [4], [5], [6]

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{y} \quad \|y_{\mathrm{d}} - \widehat{y}\| \\ &\text{subject to} \quad \operatorname{rank} \mathscr{H}_{\bar{n}+1}(\widehat{y}) \leq \bar{n}. \end{aligned} \tag{4}$$

B. New setup: structured noise

In this paper, we consider the following new setup of the signal from noise separation problem.

1) The true signal $\bar{y}$ is a trajectory of a low-complexity multiple-output linear time-invariant system.
2) The noise has two components: structured noise $\widetilde{y}$, each component $\widetilde{y}^i$ of which is a response of low-complexity linear time-invariant system and unstructured noise e, which is a zero-mean white Gaussian process.

Let N be the number of outputs. For the signal from noise separation problem in the new setup to be well-posed, we make the following assumptions:

A1: there are at least two outputs, *i.e.*, $N \geq 2$, and

A2: the models for the structured noise components $\widetilde{y}^1, \ldots, \widetilde{y}^N$ have no common poles.

In case of exact data, the signal from noise separation problem in the new setup is closely related to the greatest common divisor problem. Indeed, under the assumption that $\bar{y}^1, \ldots, \bar{y}^N$ are persistently exciting of sufficiently high order, the models $\mathscr{B}_1, \ldots, \mathscr{B}_N$ are identifiable from these signals. Then, under assumption A2, a kernel representation of the common dynamics model is given by the greatest common divisor of the kernel representations of $\mathscr{B}_1, \ldots, \mathscr{B}_N$.

Since both realization and greatest common divisor computation are classical problems for which readily available methods exist, the above observation suggests a method for solving the exact signal from noise separation problem in the new setup: 1. model the signals, 2. compute the greatest common divisor of the polynomials representing the models. Therefore, detection of the true signal from the noisy data in the new setup is a *common dynamics estimation problem*.

Apart from the signal from noise separation problem outlined above, the common dynamics estimation problem occurs in biomedical signal processing [7], monitoring of material structures [8], and audio modeling [9]. Despite of its practical applications and theoretical richness, however, the only reference we are aware of that treats the common dynamics estimation is [10]. In [10], the authors present a subspace method for common dynamics estimation, which is essentially a method for computing the intersection of subspaces. Using the behavioral approach, we show a system theoretic interpretation of the method in [10] and derive alternative subspace methods. Monte Carlo simulation study shows that some of the alternative methods derived in this paper outperform the method of [10]. We relate the methods derived to the computer algebra problem of approximate common divisor computation [11], [12], [13].

The paper is organized as follows. In Section II, we define formally the approximate common dynamics estimation problem. Optimization-based and suboptimal subspace strategies for solving the problem are presented in, respectively, Sections III and IV. The resulting methods from the subspace class are compared empirically by a Monte Carlo simulation study in Section V. The conclusions are summarized in Section VI.

II. Problem formulation

We use the behavioral approach to system theory, where a dynamical system $\mathscr{B}$ is defined as a set of trajectories. The notation $y \in \mathscr{B}$ is a convenient way of saying that y is a trajectory of $\mathscr{B}$. We denote with $\mathscr{L}_n$ the class of scalar linear time-invariant systems of order at most n. The notation $\mathscr{B} \in \mathscr{L}_n$ means that $\mathscr{B}$ is a linear time-invariant system of order at most n. $\mathscr{B}|_L$ and $y|_L$, where L is an integer, denote the restriction of, respectively, the system $\mathscr{B}$ and the signal y to the interval $1,\dots,L$.

The common dynamics estimation problem, considered in this paper, is: Given trajectories $y^1,\dots,y^N$ of systems $\mathscr{B}_1 \in \mathscr{L}_{n_1}$, …, $\mathscr{B}_N \in \mathscr{L}_{n_N}$, find the intersection $\bar{\mathscr{B}} := \mathscr{B}_1 \cap \cdots \cap \mathscr{B}_N$. We refer to $\bar{\mathscr{B}}$ as the *common dynamics* of the systems $\mathscr{B}_1,\dots,\mathscr{B}_N$. In the generic case when $\mathscr{B}_1,\dots,\mathscr{B}_N$ have no common dynamics (*i.e.*, $\mathscr{B}_1 \cap \cdots \cap \mathscr{B}_N = \{0\}$), a problem of *approximate common dynamics* estimation is considered [14].

Let σ be the *backwards shift operator*

$$(\sigma y)(t) := y(t+1), \text{ for all } t.$$

A kernel representation

$$\mathscr{B} = \ker p(\sigma) := \{\, y \mid p_0 y + p_1 \sigma y + \cdots + p_n \sigma^n y = 0 \,\}, \tag{5}$$

of the system $\mathscr{B}$ is a difference equation representation. It is defined by a scalar univariate polynomial

$$p(z) = p_0 + p_1 z + \cdots + p_n z^n.$$

If kernel representations

$$\mathscr{B}_1 = \ker p^1(\sigma), \quad \dots \quad , \mathscr{B}_N = \ker p^N(\sigma)$$

are given, then, the kernel representation $\ker p(\sigma)$ of their common dynamics $\mathscr{B}$ is given by the greatest common divisor of the N polynomials $p^1(z),\dots,p^N(z)$. The approximate common dynamics problem is then equivalent to the approximate common factor computation problem for a set of polynomials.

In this paper, we are not given the systems $\mathscr{B}_1,\dots,\mathscr{B}_N$. Instead, we are given (noisy) trajectories

$$y^i_{\mathrm{d}} = \big(y^i_{\mathrm{d}}(1),\dots,y^i_{\mathrm{d}}(T_i)\big)$$

of these systems. The problem of finding the exact or approximate common dynamics $\bar{\mathscr{B}}$ from the trajectories $y^1_{\mathrm{d}},\dots,y^N_{\mathrm{d}}$ is called the *data-driven common dynamics* estimation problem [15]. In addition, to the trajectories

$y_{\mathrm{d}}^1, \dots, y_{\mathrm{d}}^N$, we assume that we are given the dimensions $n_1, \dots, n_N$ of the models $\mathscr{B}_1, \dots, \mathscr{B}_N$ and the dimension $\bar{n}$ of their intersection $\bar{\mathscr{B}}$.

The data generation model is

$$y_{\mathrm{d}}^i = \bar{y}^i + \widetilde{y}^i + e^i, \quad \text{where} \quad \bar{y}^i \in \bar{\mathscr{B}} \in \mathscr{L}_{\bar{n}}, \quad \widetilde{y}^i \in \widetilde{\mathscr{B}}_i \in \mathscr{L}_{\widetilde{n}_i}, \quad \text{and} \quad e^i \sim \mathrm{N}(0, \sigma^2 I), \quad \text{for } i = 1, \dots, N. \tag{6}$$

where $\bar{y}^i$ is the true signal, $\widetilde{y}^i$ is the structured noise, and e is the unstructured noise. We assume that the structured noise models $\widetilde{\mathscr{B}}_1, \dots, \widetilde{\mathscr{B}}_N$ have no common poles. The problem is to estimate the true signals $\bar{y}^1, \dots, \bar{y}^N$ and the true common dynamics model $\bar{\mathscr{B}}$, given the noisy signals $y_{\mathrm{d}}^1, \dots, y_{\mathrm{d}}^N$ and the model orders $\bar{n}$, $\widetilde{n}_1$, …, $\widetilde{n}_N$. Define,

$$n_i := \bar{n} + \widetilde{n}_i.$$

The maximum-likelihood estimation problem for the estimation problem (6) is

$$\begin{aligned} &\text{minimize} \quad \sqrt{\sum_{i=1}^{N} \|y^i - \widehat{y}^i\|_2^2} \\ &\text{over } \widehat{y}^1, \dots, \widehat{y}^N \text{ and } \widehat{\mathscr{B}}, \widehat{\mathscr{B}}_1, \dots, \widehat{\mathscr{B}}_N \\ &\text{subject to} \quad \widehat{y}^i \in \widehat{\mathscr{B}}_i \in \mathscr{L}_{n_i}, \text{ for } i = 1, \dots, N \\ &\qquad\qquad\quad \text{and} \quad \widehat{\mathscr{B}} = \widehat{\mathscr{B}}_1 \cap \cdots \cap \widehat{\mathscr{B}}_N \in \mathscr{L}_{\bar{n}}. \end{aligned} \tag{7}$$

III. Maximum-likelihood estimation strategies

In the classical setup of the signal from noise estimation problem, the maximum-likelihood estimation problem is equivalent to a structured low-rank approximation problem. Similarly, we show in Section III-A that in the new setup the maximum-likelihood estimation problem is equivalent to a structured low-rank approximation problem. However, the latter problem formulation has multiple rank constraints. Two alternative strategies for solving the maximum-likelihood estimation problem, shown in Sections III-B and III-C, are based on structured kernel and state-space representations.

A. Reformulation as structured matrix approximation with multiple rank constraints

We denote by span$A$ the span of the columns of the matrix A. Under the persistency of excitation assumption A2, we have that

$$\mathscr{B}_i|_L = \operatorname{span} \mathscr{H}_L(\bar{y}^i), \qquad \text{for } i = 1, \dots, N. \tag{8}$$

Then, the common dynamics $\bar{\mathscr{B}} := \mathscr{B}_1 \cap \cdots \cap \mathscr{B}_N$ is given by

$$\bar{\mathscr{B}}|_L = \operatorname{span} \underbrace{\begin{bmatrix} \mathscr{H}_L(\bar{y}^1) & \cdots & \mathscr{H}_L(\bar{y}^N) \end{bmatrix}}_{\mathscr{H}_L(\bar{y})}.$$

The matrix $\mathscr{H}_L(\bar{y})$ is called *mosaic-Hankel* [16]. The final element that we need for the link between the maximum-likelihood estimation problem (7) and low-rank approximation is the equivalence between trajectories of linear time-invariant systems of bounded complexity and rank deficient Hankel matrices

$$y \in \mathscr{B} \in \mathscr{L}_n \quad \iff \quad \operatorname{rank} \mathscr{H}_L(y) \leq n.$$

Theorem 1. *The maximum-likelihood estimation problem (7) is equivalent to the low-rank approximation problem*

$$\begin{aligned} &\textit{minimize} \quad \textit{over } \widehat{y}^1,\ \ldots,\ \textit{and } \widehat{y}^N \quad \sqrt{\sum_{i=1}^{N} \|y^i - \widehat{y}^i\|_2^2} \\ &\textit{subject to} \quad \operatorname{rank} \mathscr{H}_{n_i+1}(\widehat{y}^i) \leq n_i,\ \textit{for } i = 1,\ldots,N \\ &\qquad\quad \textit{and} \quad \operatorname{rank} \mathscr{H}_{\bar{n}+1}(\widehat{y}) \leq \bar{n}. \end{aligned}$$

Solution methods for structured low-rank approximation problems with multiple rank constraints are currently under development.

B. Reformulation based on structured kernel representation

Let $\ker p^i(\sigma)$ be a minimal kernel representation of the model $\mathscr{B}_i$ and $\ker \bar{p}(\sigma)$ be a minimal kernel representation of the model $\bar{\mathscr{B}}$. Since $\bar{p}(z)$ is a common factor of $p^1(z),\ldots,p^N(z)$, there are polynomials $\widetilde{p}^i(z)$, such that

$$p^i(z) = \widetilde{p}^i(z)\bar{p}(z).$$

Moreover, by the assumption that the kernel representations are minimal and by assumption A2, $\bar{p}(z)$ is the greatest common factor of $p^1(z),\ldots,p^N(z)$.

The product of polynomials $p^i(z) = \widetilde{p}^i(z)\bar{p}(z)$ corresponds to convolution of their coefficients vectors $p^i = \widetilde{p}^i \star \bar{p}$. (With some abuse of notation, in what follows, we use the same notation $\bar{p}(z)$ *etc.* for the polynomial $\bar{p}_0 + z\bar{p}_1 + \cdots + z^{\bar{n}}\bar{p}_{\bar{n}}$ and for the vector $\bar{p} = \begin{bmatrix} \bar{p}_0 & \bar{p}_1 & \cdots & \bar{p}_{\bar{n}} \end{bmatrix}^\top$ of its coefficients.) The following result gives equivalent optimization problems to (7) based on the kernel representations of the model.

Theorem 2. *Problem (7) is equivalent to the following problem:*

$$\begin{aligned} &\textit{minimize} \quad \textit{over } \widehat{y}^i,\ \widehat{\widetilde{p}}^i,\ \textit{and } \widehat{p} \quad \|y - \widehat{y}\|_2 \\ &\textit{subject to} \quad (\widehat{\widetilde{p}}^i \star \widehat{p})\mathscr{H}_{n_i+1}(\widehat{y}^i) = 0, \quad \textit{for } i = 1,\ldots,N. \end{aligned} \tag{9}$$

Problem (9), in turn, is equivalent to a structured low-rank approximation problem with a bi-linear constraint on the kernel parameter. Currently, there are no efficient solution methods for this class of structured low-rank approximation problems.

C. Reformulation based on structured state-space representation

An autonomous linear time-invariant system admits a minimal state-space representation

$$\mathscr{B}(A,C) := \{\, y \mid \text{there is } x, \text{ such that } \sigma x = Ax,\ y = Cx \,\}.$$

With some loss of generality, we assume that the system can be represented in the modal form, where A is a diagonal matrix. The diagonal elements of A are the poles of the system. The common dynamics model implies existence of $\bar{n}$ common poles $\bar{\mu}_1,\ldots,\bar{\mu}_{\bar{n}}$. Let $\bar{\mu} \in \mathbb{C}^{\bar{n}}$ be the vector of common poles. Each mode $\mathscr{B}_i$ has $\widetilde{n}_i = n_i - \bar{n}$ additional poles $\lambda_1^i,\ldots,\lambda_{\widetilde{n}_i}^i$. Let $\lambda^i \in \mathbb{C}^{\widetilde{n}_i}$ be the vector of additional poles for $\mathscr{B}_i$.

The model with output $\bar{y} = \begin{bmatrix} \bar{y}^1 & \cdots & \bar{y}^N \end{bmatrix}^\top$ is given by a structured state-space representation

$$\bar{y} \in \mathscr{B}\Big(\operatorname{diag}(\lambda^1, \dots, \lambda^N, \bar{\mu}), \begin{bmatrix} c_1 & & & c_1' \\ & \ddots & & \vdots \\ & & c_N & c_N' \end{bmatrix}\Big),$$

where $c_i \in \mathbb{C}^{1 \times \tilde{n}_i}$, $c_i' \in \mathbb{R}^{1 \times \bar{n}}$ are model parameters. The structure of the C matrix imposes the constraint that the last $\bar{n}$ poles are common poles for all outputs. This leads to the following result.

Theorem 3. *Problem (7) is equivalent to the following problem:*

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{y}^i,\ \widehat{\lambda}^i,\ \widehat{\mu},\ \widehat{c}_i, \text{ and } \widehat{c}_i' \quad \|y - \widehat{y}\|_2 \\ &\text{subject to} \quad \widehat{y} \in \mathscr{B}\Big(\operatorname{diag}(\widehat{\lambda}^1, \dots, \widehat{\lambda}^N, \widehat{\mu}), \\ &\qquad\qquad \begin{bmatrix} \widehat{c}_1 & & & \widehat{c}_1' \\ & \ddots & & \vdots \\ & & \widehat{c}_N & \widehat{c}_N' \end{bmatrix}\Big). \end{aligned} \tag{10}$$

Problem (10) is a gray-box identification problem [17].

IV. Subspace methods

The subspace methods are based on the fact that, under the persistency of excitation assumption, a restriction of the model is given by the span of a Hankel matrix constructed from the data (8). Then, the common subspace $\bar{\mathscr{B}}$ can be computed as an intersection of subspaces

$$\bar{\mathscr{B}}|_L = \mathscr{B}_1|_L \cap \cdots \cap \mathscr{B}_N|_L. \tag{11}$$

The final step of the subspace algorithms is parameter estimation — computing a representation of $\bar{\mathscr{B}}$.

To summarize, the subspace methods for identification of common dynamics presented in this section have the following main steps:

1) **modeling/preprocessing**

   compute models $\mathscr{B}_1, \dots, \mathscr{B}_N$ for the signals $y_\mathrm{d}^1, \dots, y_\mathrm{d}^N$, using the prior information that $\mathscr{B}_i \in \mathscr{L}_{n_i}$,

2) **subspace intersection**

   compute $\bar{\mathscr{B}} = \mathscr{B}_1 \cap \cdots \cap \mathscr{B}_N$, and

3) **parameter estimation**

   compute a representation of $\bar{\mathscr{B}}$.

A. Modeling/preprocessing

In the case of exact data, the models $\mathscr{B}_1, \dots, \mathscr{B}_N$ are given directly by the spans of the Hankel matrices $\mathscr{H}_L(y_\mathrm{d}^1), \dots, \mathscr{H}_L(y_\mathrm{d}^N)$, see (8). In case of the inexact (noisy) data, the prior knowledge that the model orders are n_1, …, n_N is used, *i.e.*, the dimensions of $\mathscr{B}_1$, …, $\mathscr{B}_N$ are restricted to n_1, …, n_N, respectively. The problem of computing an optimal models in case of inexact data (2) is a Hankel low-rank approximation problem, see (4). The Hankel low-rank approximation problem, however, is a nonconvex optimization problem. Subspace methods use suboptimal heuristics.

A common suboptimal heuristic for solving the data modeling problem (2) is to ignore the structure and do unstructured low-rank approximation of the Hankel matrix $\mathscr{H}_L(y_{\mathrm{d}}^i)$. The rank is enforced to be n_i by truncation of the singular value decomposition. The low-rank approximation by truncation of the singular value decomposition to the n_i dominant ones can be viewed as a data preprocessing step. Subsequently state-space parameters of an approximate model can be estimated from the low-rank approximation by solving approximately an overdetermined system of linear equations by the ordinary least-squares method. The resulting method for solving (2) is known as Kung's method [18], [6].

In [10], a modification of Kung's method is used—the total least squares method [19] is used for the parameter estimation rather than the ordinary least square method. The simulation results, shown in Section V, suggest that this modification of Kung's method leads to worse performance of the overall common dynamics estimation method.

```
function [sys, yh, xini, err] = y02ss(y, n)

if length(size(y)) == 2
    [p, T] = size(y);
    if p > T
        y = [zeros(1, T); y];
    else
        y = [zeros(p, 1), y];
    end
else
    error('3d y not implemented yet')
end

[sys, yh, O, C, err] = h2ss(y, n);
yh = yh(2:end, :); xini = sys.b;
sys = ss(sys.a, [], sys.c, [], -1);
```

B. *Subspace intersection*

The second step of the common dynamics estimation methods is computing of the subspace intersection (11). General methods for subspace intersection based on image and kernel representation are presented in the Appendix, see also [20, Section 12.4]. The subspaces $\mathscr{B}_1, \dots, \mathscr{B}_N$ have the special property that they are behaviors of autonomous linear time-invariant systems. Exploiting this special structure allows us to develop more efficient methods.

One approach of exploiting the special structure of $\mathscr{B}_1,\dots,\mathscr{B}_N$ is to compute parameters $p^1(z),\dots,p^N(z)$ of kernel representations $\ker p^1(\sigma)$, $\dots$, $\ker p^N(\sigma)$. Then, the subspace intersection problem becomes a problem of computing the greatest common divisor of a set of polynomials. There are existing methods for greatest common factor computation. In case of inexact data, generically, an exact common factor does not exist. In this case, the aim is to find an approximate common factor of degree $\bar{n}$. Again, this is a well developed problem in computational algebra.

Computing an approximate greatest common factor however is a nonconvex optimization problem. In fact, as shown in [21] this problem is also a structured low-rank approximation problem. This motivates an alternative suboptimal method that first computes an approximate intersection $\mathscr{B}$ using one of the general methods presented in the Appendix and then models the resulting subspace as a linear time-invariant behavior.

```
function [Ch, Ph, f] = alcf_ss(P, m, d)
dP = size(P, 1) / m - 1;
dA = dP - d; T = 2 * dP + 1;
Mp = multmat(P, m, T - dP - 1);
Y = lra(Mp, size(Mp, 1) - d)';
for i = 1:d,
  w{i} = reshape(Y(:, i), m, T);
end
Ch = lra(moshank(w, d + 1), d * m)';
Ch = Ch / Ch(1:m, :);
[f, Ph] = costfun(Ch(m+1:end, :), P, m, dA);
```

```
function [f, Ph] = costfun(C, P, m, dA)
C = [eye(size(C, 2)); C];
Mc = multmat(C, m, dA);
Ph = Mc * (Mc \ P);
f = norm(P - Ph, 'fro');
```

C. *Parameter estimation*

The final step of the subspace method is estimation of the parameters of the estimated common subspace $\widehat{\mathscr{B}}$. With exact data, by construction, $\widehat{\mathscr{B}} = \bar{\mathscr{B}} \in \mathscr{L}_{\bar{n}}$. In this case, the state-space parameter estimation problem is the classical realization problem. With inexact data, however, generically, $\widehat{\mathscr{B}} \notin \mathscr{L}_{\bar{n}}$. In this case an approximation $\widehat{\mathscr{B}}'$ of $\widehat{\mathscr{B}}$ is computed, such that $\widehat{\mathscr{B}}' \in \mathscr{L}_{\bar{n}}$ and $\widehat{\mathscr{B}}'$ is as close as possible to $\widehat{\mathscr{B}}$.

The parameter estimation problem with inexact data is also a system identification problem, however, for multiple trajectories. Let $z^1,\dots,z^{\bar{n}}$ form a basis for $\widehat{\mathscr{B}}$. We aim to fit $z^1,\dots,z^{\bar{n}}$ by a linear time-invariant system in the model class $\mathscr{L}_{\bar{n}}$. This problem can be solved again by Kung's method.

The literate programming [22], [23] implementation of the methods is given in a full version of this paper [24].

V. Simulation examples

In this section, we compare the following subspace methods for common dynamics estimation:

1) `papy` — the original method of [10],
2) `papy-ls` — the method of [10] with ordinary least squares instead of total least squares parameter estimation,
3) `gv` — the subspace method presented in Section IV,
4) `raw` — the subspace method presented in Section IV without preprocessing, and
5) `alcf` — data modeling followed by approximate common factor computation [25].

The simulation setup is (6) with $N = 2$ signals, generated by systems of orders $n_1 = n_2 = 8$, and $\bar{n} = 4$ common poles. The signal lengths are $T_1 = T_2 = 150$ and the noise standard deviation is $\sigma = 0.25$. The true models $\bar{\mathscr{B}}_1$ and $\bar{\mathscr{B}}_2$ are randomly generated marginally stable systems.

Let $\bar{p}(z)$ be a monic polynomial that defines a minimal kernel representation $\ker \bar{p}(\sigma)$ of the true common dynamics $\bar{\mathscr{B}}$ and let $\widehat{p}(z)$ be a monic polynomial that defines a minimal kernel representation $\ker \widehat{p}(\sigma)$ of the estimated common dynamics $\widehat{\mathscr{B}}$. We define the following estimation errors

$$e_p = \frac{\|\bar{p} - \widehat{p}\|}{\|\bar{p}\|} \qquad \text{and} \qquad e_y = \frac{\|\bar{y} - \widehat{y}\|}{\|\bar{y}\|}$$

for the comparison of the methods: e_p is the relative *parameter error* and e_y is the relative *signal error*. The results of a Monte-Carlo simulation with 100 repetitions are shown in Table I.

method	e_p	y_y
`papy`	0.2460	0.2053
`papy-ls`	0.2182	0.2084
`gv`	0.2182	0.2084
`raw`	0.3468	0.3163
`alcf`	0.1368	0.1868

TABLE I

The results from a Monte-Carlo simulation, comparing the subspace methods for common dynamics estimation, show that `papy-ls` is equivalent to `gv`, using ordinary least squares (`papy-ls`) gives better performance than total least squares (`papy`), no preprocessing (`raw`) gives worse performance, and exploiting the linear time-invariant structure in the subspace intersection (`alcf`) improves the performance.

```
clear all

% simulation parameters
T = 150; t = (1:T)';
```

```
k = [2 2]; kc = 2;
n = 2 * k; nc = 2 * kc;
s = 0.25; N = 100;

w = rand(1, kc + sum(k));
phi = rand(1, kc + sum(k));
Y = sin(2 * pi * t * w + phi(ones(T, 1), :));
y(:, 1) = Y(:, 1:(kc + k(1))) * rand(kc + k(1), 1);
y(:, 2) = Y(:, [1:kc (kc + k(1) + 1):(kc + sum(k))]) * rand(kc + k(2), 1);
% plot(y, '-'), hold on, plot(yd, '--')

sys1 = y02ss(y(:, 1), n(1) + nc);
sys2 = y02ss(y(:, 2), n(2) + nc);
[ch, yh] = common_dynamics_alcf(y, n, nc);
norm(y(:) - yh(:)) / norm(y(:))
pc = poly(roots(ch))';

%% methods
opt{1}.m = 'papy'; opt{1}.tls = 1;
opt{2}.m = 'papy'; opt{2}.tls = 0;
opt{3}.m = 'gv'  ; opt{3}.tls = 0;
opt{4}.m = 'raw' ; opt{4}.tls = 0;
opt{5}.m = 'alcf'; opt{5}.tls = 0;

%% MC simulation
for i = 1:N
  yt = randn(size(y));
  yd = y + s * yt / norm(yt(:)) * norm(y(:));
  for j = 1:length(opt)
    [ch, yh] = common_dynamics(yd, n, nc, opt{j});
    Ph{j}(:, i) = poly(roots(ch))';
    Yh{j}(:, i) = yh(:);
  end
end

%% print results
```

```
vec_y = y(:); Y = vec_y(:, ones(1, N));
Pc = pc(:, ones(1, N));
for j = 1:length(opt)
  res(j, :) = [norm(Y - Yh{j}, 'fro') / norm(Y), ...
               norm(Pc - Ph{j}, 'fro') / norm(Pc)];
end

[{'' 'Ey' 'Ez'}; ...
 [{'papy'; 'papy-ls'; 'gv'; 'raw'; 'alcf'}, ...
 num2cell(res)]]
```

VI. Conclusions

We motivated the need to estimate common dynamics estimation among several observed signals by a new signal from noise separation problem. In the new problem the noise has two components—structured noise, which is a trajectory of a low-complexity linear time-invariant system and unstructured noise, which is a zero-mean white Gaussian process. The problem is well-posed when the structured noise models of the observed signals have no common poles.

Two solution approaches for the the common dynamics estimation problem were considered:

1) maximum likelihood estimation, and
2) subspace estimation.

The maximum likelihood estimation problem is equivalent to a structured low-rank approximation problem with multiple rank constraints. Alternatively, the maximum likelihood estimation problem can be solved by using structured kernel and state-space representations. However, currently, there are no efficient optimization methods available to solve the maximum likelihood estimation problem.

We developed a generic subspace estimation method that has the following steps:

1) modeling of the observed signals, which serves a preprocessing role,
2) computation of the intersection of the models obtained in step 1, and
3) parameter estimation of the common subspace, which yields the parameter of interest — the common poles.

Simulation results comparing five variations of the generic subspace method suggest improvements of previously proposed subspace methods in the literature [10]. First, parameter estimation by the ordinary least squares method yields more accurate results in parameter estimation error than parameter estimation by the total least squares method. Second, exploiting the linear-time invariant structure in the subspace intersection step by using methods for approximate common factor computation further improves the estimation accuracy.

VII. Acknowledgements

The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC Grant agreement number 258581 "Structured low-rank approximation: Theory, algorithms, and applications" and Fund for Scientific Research Vlaanderen (FWO) projects G028015N "Decoupling multivariate polynomials in nonlinear system identification" and G090117N "Block-oriented nonlinear identification using Volterra series"; and Fonds de la Recherche Scientifique (FNRS) – FWO Vlaanderen under Excellence of Science (EOS) Project no 30468160 "Structured low-rank matrix / tensor approximation: numerical optimization-based algorithms and applications".

References

[1] J. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory*. New York: Springer-Verlag, 1998.

[2] J. C. Willems, "The behavioral approach to open and interconnected systems: Modeling by tearing, zooming, and linking," *Control Systems Magazine*, vol. 27, pp. 46–99, 2007.

[3] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor, *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. SIAM, 2006.

[4] I. Markovsky, "Structured low-rank approximation and its applications," *Automatica*, vol. 44, no. 4, pp. 891–909, 2008.

[5] ——, "Recent progress on variable projection methods for structured low-rank approximation," *Signal Processing*, vol. 96PB, pp. 406–419, 2014.

[6] ——, *Low-Rank Approximation: Algorithms, Implementation, Applications*. Springer, 2019.

[7] S. Van Huffel, "Enhanced resolution based on minimum variance estimation and exponential data modeling," *Signal processing*, vol. 33, no. 3, pp. 333–355, 1993.

[8] L. Rippert, "Optical fibers for damage monitoring in carbon fiber reinforced plastic composite materials," Ph.D. dissertation, Katholieke Universiteit Leuven, 2005.

[9] R. Boyer and K. Abed-Meraim, "Audio modeling based on delayed sinusoids," *IEEE Transactions on speech and audio processing*, vol. 12, no. 2, pp. 110–120, 2004.

[10] J.-M. Papy, L. De Lathauwer, and S. Van Huffel, "Common pole estimation in multi-channel exponential data modeling," *Signal Processing*, vol. 86, no. 4, pp. 846–858, Apr. 2006.

[11] N. Karmarkar and Y. Lakshman, "On approximate GCDs of univariate polynomials," in *J. Symbolic Comput.*, S. Watt and H. Stetter, Eds., vol. 26, 1998, pp. 653–666, special issue on Symbolic Numeric Algebra for Polynomials.

[12] E. Kaltofen, R. M. Corless, and D. J. Jeffrey, "Challenges of symbolic computation: my favorite open problems," *J. Symbolic Comput.*, vol. 29, no. 6, pp. 891–919, 2000.

[13] Z. Zeng, "The approximate gcd of inexact polynomials. part i: A univariate algorithm," http://homepages.neiu.edu/~naclab/, 2004.

[14] I. Markovsky and N. Guglielmi, "Model order estimation based on a method for computing distance to uncontrollability," in *Proc. of the Conference on Noise and Vibration Engineering (ISMA)*, Leuven, Belgium, September 2016, pp. 2963–2970.

[15] I. Markovsky, A. Fazzi, and N. Guglielmi, "Applications of polynomial common factor computation in signal processing," in *Latent Variable Analysis and Signal Separation*, ser. Lecture Notes in Computer Science. Springer, 2018, pp. 99–106.

[16] G. Heinig, "Generalized inverses of Hankel and Toeplitz mosaic matrices," *Linear Algebra Appl.*, vol. 216, no. 0, pp. 43–59, Feb. 1995.

[17] L. Ljung, *System Identification: Theory for the User*. Prentice-Hall, 1999.

[18] S. Kung, "A new identification method and model reduction algorithm via singular value decomposition," in *Proc. 12th Asilomar Conf. Circuits, Systems, Comp.*, 1978, pp. 705–714.

[19] I. Markovsky and S. Van Huffel, "Overview of total least squares methods," *Signal Proc.*, vol. 87, pp. 2283–2302, 2007.

[20] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.

[21] I. Markovsky and S. Van Huffel, "An algorithm for approximate common divisor computation," in *Proc. 17th Symp. on Math. Theory of Networks and Systems*, Kyoto, Japan, 2006, pp. 274–279.

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

[23] N. Ramsey, "Literate programming simplified," *IEEE Software*, vol. 11, pp. 97–105, 1994.

[24] I. Markovsky, T. Liu, and A. Takeda, "Common dynamics estimation," Dept. ELEC, Vrije Universiteit Brussel, Tech. Rep., 2019. [Online]. Available: http://homepages.vub.ac.be/~imarkovs/publications/common-dynamics.pdf

[25] W. Qiu, Y. Hua, and K. Abed-Meraim, "A subspace method for the computation of the GCD of polynomials," *Automatica*, vol. 33, no. 4, pp. 741–743, 1997.

VIII. Appendix

We consider the subspace intersection problem: given subspaces

$$\mathscr{B}_i = \operatorname{image} P_i = \ker R_i \subset \mathbb{R}^q, \quad n_i = \dim \mathscr{B}_i, \ \text{for } i = 1, \dots, N,$$

find their intersection

$$\mathscr{B} := \mathscr{B}_1 \cap \cdots \cap \mathscr{B}_N = \operatorname{image} P = \ker R \subset \mathbb{R}^q, \quad n = \dim \mathscr{B}.$$

There are two versions of the problem

1) exact intersection, and
2) approximate intersection, with specification of the dimension n of the common subspace.

A. Kernel representation

The matrix

$$R = \begin{bmatrix} R_1 \\ \vdots \\ R_N \end{bmatrix}$$

defines an exact (nonminimal) kernel representation of the intersection $\mathscr{B}$. Computing a minimal representation R' requires finding a nonsingular matrix U, such that

$$UR = \begin{bmatrix} R' \\ 0 \end{bmatrix}, \qquad \text{with } R' \text{ full row rank.}$$

Computing a kernel representation of an approximate intersection with dimension n requires a rank-$(q-n)$ approximation of R.

B. Image representation

Next, we derive an image representation of the common subspace $\mathscr{B}$. We have that

$$\dim \operatorname{image} \underbrace{\begin{bmatrix} P_1 & \cdots & P_N \end{bmatrix}}_{P} = n_1 + \cdots + n_N - (N-1)n,$$

so that there is an $(n_1 + \cdots + n_N) \times (N-1)n$ matrix Z, such that

$$\begin{bmatrix} P_1 & \cdots & P_N \end{bmatrix} \begin{bmatrix} Z_1 \\ \vdots \\ Z_N \end{bmatrix} = 0.$$

Note that

$$\mathrm{span}(P_1 Z_1) = \cdots = \mathrm{span}(P_N Z_N) = \mathscr{B}.$$

Therefore, a parameter P of an image representation of $\mathscr{B}$ is given by

$$P = P_1 Z_1.$$

Computing an image representation of an approximate intersection with dimension n requires a rank

$$(n_1 + \cdots + n_N - (N-1)n)$$

approximation of P.

IX. Implementation of the methods

A. Using kernel representation and approximate common divisor computation

```
function [ch, yh] = common_dynamics_alcf(y, n, nc)
% step 1
r1 = poly(eig(y02ss(y(:, 1), n(1) + nc)));
r2 = poly(eig(y02ss(y(:, 2), n(2) + nc)));
% step 2
[ch, rh] = alcf_ss(collect(r1', r2'), 1, nc);
% step 3
yh(:, 1) = ry2yh(rh(:, 1)', y(:, 1));
yh(:, 2) = ry2yh(rh(:, 2)', y(:, 2));
function c = collect(a, b)
c = [[zeros(size(b, 1) - size(a, 1), 1); a] ...
     [zeros(size(a, 1) - size(b, 1), 1); b]];
```

B. Using image representation and various methods for computing intersection of subspaces

```
function [ch, yh] = common_dynamics(y, n, nc, opt)
% x -> y, K -> n, G = 2
N = size(y, 1); L = round(N / 2); M = N - L + 1;
if ~exist('opt'), opt.m = 'papy'; opt.tls = 1; end
% step 1
H1 = blkhank(y(:, 1), L); [r1, p1] = lra(H1, n(1) + nc);
H2 = blkhank(y(:, 2), L); [r2, p2] = lra(H2, n(2) + nc);
<<image -> kernel>>
% step 2
switch lower(opt.m)
```

```
  case ’papy’
    [rc, pc] = lra([p1 p2], nc);
    <<pc -> ch>>
  case ’gv’
    [rc, ~, p1p2h] = lra([p1 p2]’, n(1) + n(2) + nc);
    pc = p1p2h(1:n(1) + nc, :)’ * rc(:, 1:n(1) + nc)’;
    <<pc -> ch>>
  case ’raw’
    j = N - max(n) - nc + 1;
    H1 = blkhank(y(:, 1), n(1) + nc, j);
    H2 = blkhank(y(:, 2), n(2) + nc, j);
    [rc, ~, p1p2h] = lra([H1; H2], n(1) + n(2) + nc);
    pc = p1p2h(1:n(1) + nc, :)’ * rc(:, 1:n(1) + nc)’;
    <<pc -> ch>>
  case ’alcf’
    [ch, rh] = alcf_ss(collect(r1’, r2’), 1, nc);
  otherwise
    disp(’Unknown method.’)
end
% step 3
if nargout > 1
  if ~exist(’rh’)
    [~, rh] = costfun(ch(2:end)’, collect(r1’, r2’), 1, max(n));
  end
  yh(:, 1) = ry2yh(rh(:, 1)’, y(:, 1));
  yh(:, 2) = ry2yh(rh(:, 2)’, y(:, 2));
end
```

image -> kernel

```
if ~isfield(opt, ’tls’) || ~opt.tls
  r1 = poly(eig(p1(1:end-1, :) \ p1(2:end, :)));
  r2 = poly(eig(p2(1:end-1, :) \ p2(2:end, :)));
else
  r1 = poly(eig(tls(p1(1:end-1, :), p1(2:end, :))));
  r2 = poly(eig(tls(p2(1:end-1, :), p2(2:end, :))));
end
```

pc -> ch

```
if ~isfield(opt, 'tls') || ~opt.tls
  ch = poly(eig(pc(1:end-1, :) \ pc(2:end, :)));
else
  ch = poly(eig(tls(pc(1:end-1, :), pc(2:end, :))));
end
```

C. *Computing the approximation signals*

In [10], the approximations $\widehat{y}^1, \ldots, \widehat{y}^N$ of $y^1, \ldots, y^N$ are not computed. Once the signals are modeled as described above, $\widehat{y}^1, \ldots, \widehat{y}^N$ can be computed using the models and the data $y^1, \ldots, y^N$. This is a Kalman smoothing problem.

```
function yh = ry2yh(r, y)
T = length(y);
R = multmat(fliplr(r)', 1, T - length(r))'; P = null(R);
yh = P * (P \ y);
```