Automatica 139 (2022) 110214

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Cepstral identification of autoregressive systems[☆]

Oliver Lauwers [*], Christof Vermeersch, Bart De Moor [1]

Center for Dynamical Systems, Signal Processing, and Data Analytics (STADIUS), Department of Electrical Engineering (ESAT), KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium

ARTICLE INFO

Article history:
Received 30 June 2020
Received in revised form 2 December 2021
Accepted 13 January 2022
Available online 5 March 2022

Keywords:
System identification
Signal processing
Autoregressive systems
Discrete-time systems
Difference equations
Polynomials
Classification

ABSTRACT

Recent research has provided a better understanding of the power cepstrum, which has led to several applications in time series clustering, classification, and anomaly detection. It has also provided a deeper understanding of the theoretical framework that relates the power cepstrum with some system theoretic properties of the underlying dynamics. In this paper, we pursue the intricate connections between the power cepstrum of a signal and the pole polynomial of the underlying generative model. In this way, we develop a simple and extremely efficient method to identify an autoregressive (AR) system, starting from the power cepstrum of its output signal. This general framework uses Newton's identities to set up a system of elementary symmetric polynomials over the cepstral coefficients and results in an identification algorithm that is independent of the length of the power cepstrum, with computational complexity only linearly dependent on the order of the model. We provide several numerical examples, first on synthetic time series, then on the classical Yule sunspot numbers modeling problem, and finally on a contemporary application involving structural health monitoring. Subsequently, the novel system identification algorithm is employed to provide insight in the results of weighted cepstral clustering, showing that the model estimated from the center of a cluster provides a good estimator for the dynamics in that cluster.

1. Introduction

Given the steady increase in sensor data (Perera, Zaslavsky, Christen, & Georgakopoulos, 2014), computational capacity, and storage capabilities, interest in the analysis of large time series has grown substantially over the past decades. While scaleable data-driven algorithms for pattern recognition exist, dynamical system modeling on signals remains a laborious and time-consuming task, often requiring expert knowledge. Existing system identification techniques (for a good overview of core concepts, see Ljung, 2010) do not always scale well for large data sets of long signals, though an important line of research based on the covariance extension method (see Mari, Dahlén, & Lindquist, 2000) already mitigates this problem and does feature high scalability. Novel system identification techniques offering low computational complexity are always of interest.

Various novel technologies, such as smart meters (Uddin, Ahmad, Qamar, & Altaf, 2018), wearables (Kim, Campbell, Esteban-Fernández de Ávila, & Wang, 2019), and autonomous vehicles (Edwards, 2019), only increase the need for this type of new system identification techniques. Existing approaches often apply black-box machine learning methods, but do not provide insight in the underlying dynamics of the problem. Results from these algorithms lack interpretability, which can be detrimental in an engineering context. Generating insight in the results and transparency in the method of operation should, therefore, be an important aspect of data analysis in general, and especially in time series analysis. Unsupervised learning, in particular, would benefit tremendously from methods that can be explicitly linked to the underlying dynamics or other fundamental properties of the signals, as the lack of labels or clear learning goals can easily result in confusing, uninterpretable, and sub-par results (Vellido, Martín-Guerrero, & Lisboa, 2012). Time series clustering is a prime example of such an unsupervised learning problem where results from black-box techniques are opaque. Similarity measures play a crucial role in providing interpretability, whereas traditional methods come up short (Pereira & de Mello, 2013).

Earlier work (De Cock & De Moor, 2002; Lauwers & De Moor, 2017) has mitigated this problem somewhat by proposing and extending a data-based distance measure, i.e., the weighted cepstral distance, that mimics a model norm. While this provides some

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Cristian R. Rojas under the direction of Editor Torsten Söderström.

[*] Corresponding author.
E-mail addresses: oliver.lauwers@esat.kuleuven.be (O. Lauwers), christof.vermeersch@esat.kuleuven.be (C. Vermeersch), bart.demoor@esat.kuleuven.be (B. De Moor).

[1] Fellow IEEE, SIAM.

https://doi.org/10.1016/j.automatica.2022.110214
0005-1098/© 2022 Elsevier Ltd. All rights reserved.

insight in what a clustering algorithm considers to be similar time series, it still fails to explicitly show what the resulting clusters represent. In this paper, we address this issue and develop a new system identification method that provides this insight. Our main contributions are

- introducing an extremely efficient, novel system identification technique that relies on exact solutions of a system of multivariate polynomial equations, which relates the power cepstrum of a signal to the coefficients of the pole polynomial of its generating linear model; when the power cepstrum is known, the algorithm is independent of the length of the power cepstrum and its computational complexity is only linearly dependent on the order of the model (starting from a signal with unknown power cepstrum, on the other hand, the computation of the power cepstrum dominates and the procedure has complexity $\mathcal{O}(N \log N)$, with N the signal length);
- interpreting the cluster centers from a cepstral clustering problem as good estimators for the average dynamics of the signals present in these clusters;
- exploring synthetic, historical, and contemporary illustrations and applications to corroborate the theoretical results of this paper with practical examples.

Note that the method presented in this text shares its computational efficiency with covariance extension methods (Mari et al., 2000), providing an exact solution, with a computational complexity linearly dependent on the order of the model, but now starting from the cepstrum rather than from the spectrum of the signal.

The remainder of this paper proceeds as follows: Section 2 introduces notation and model class assumptions, and it explicitly defines the identification objective that we meet in this paper. Section 3 provides relevant elements from the theory of symmetric polynomials, which are used in Section 4 to develop the new system identification algorithm. In Section 5, we give some background on cepstral clustering and show how the identification algorithm can be used to identify the representative dynamics of a cluster. Section 6 provides several numerical examples, showing the theoretical results in practice. We show results on synthetic signals, the classical Yule sunspot numbers time series, and structural health monitoring data. All code used in the numerical examples is available on GitHub.[2] We conclude this paper and introduce some ideas for future work in Section 7. Appendix A shows how the input white noise hypothesis can be relaxed, Appendix B relaxes the assumption of stable models, and Appendix C elaborates on the computational aspects of the power cepstrum.

2. Notation & objective

2.1. Model class assumptions

Consider an nth order autoregressive (AR) model, which can be represented in the z-domain as

$$Y(z) = \frac{1}{a(z)} U(z), \tag{1}$$

where $Y(z)$ is the z-transform of the output signal, $y(t)$, $U(z)$ is the z-transform of the white noise input signal, $u(t)$, and $a(z)$ is the pole polynomial of the model,

$$a(z) = z^n + \sum_{i=1}^{n} a_i z^{n-i} = \prod_{j=1}^{n} \left(z - \alpha_j\right), \tag{2}$$

with a_i the coefficients of the pole polynomial and α_j the (complex) roots of the pole polynomial, which are the poles of the AR model. We also assume $\left|\alpha_j\right| < 1$, though we relax this stability assumption in Appendix B.

We call $H(z) = \frac{1}{a(z)}$ the transfer function of the AR model and define its power spectral density, $\Phi_h\left(e^{i\omega}\right)$, as

$$\Phi_h\left(e^{i\omega}\right) = H\left(e^{i\omega}\right)\overline{H\left(e^{i\omega}\right)} = \left|H\left(e^{i\omega}\right)\right|^2, \tag{3}$$

where $\overline{\cdot}$ corresponds to the complex conjugate.

When we denote the Fourier transform as $\mathcal{F}$, the *model (power)*[3] *cepstral coefficients*, $c_h(k)$, with k the coefficient number, of a model with transfer function $H(z)$ are

$$\mathcal{F}\left\{c_h(k)\right\} = \log\left|H\left(e^{i\omega}\right)\right|^2 = \log\left(\Phi_h\left(e^{i\omega}\right)\right). \tag{4}$$

Similarly, we define the *(power) cepstrum* of a signal $y(t)$ as

$$\mathcal{F}\left\{c_y(k)\right\} = \log\left|\mathcal{F}\left\{y(t)\right\}\right|^2.$$

Computationally, we can employ the fast Fourier transform (FFT), as shown in Fig. 1, to obtain these coefficients. A detailed discussion of the computational aspects can be found in Appendix C.

In the remainder of this paper, we work with the model cepstral coefficients $c_h(k)$. When we assume white noise inputs in Eq. (1), we have that $c_y(k) = c_h(k)$ (Oppenheim & Schafer, 1975). However, we can even relax this assumption: as long as the input signal is stationary, so that its Fourier transform is well-defined, we can find the model cepstrum using the results from Appendix A.

Combining Eqs. (1), (2), and (4), we can relate the model cepstrum with the poles of the AR model as in Oppenheim and Schafer (1975):

$$c_h(k) = \sum_{j=1}^{n} \frac{\alpha_j^{|k|}}{|k|}, \quad \forall k \neq 0, \tag{5}$$

and $c_h(0) = g'$, a constant that depends on the gain and initial conditions of the system.

2.2. Identification objective

We pursue the following identification objective:

> Identify the pole polynomial, $a(z)$, from a discrete-time model cepstrum, $c_h(k)$, generated by a linear time-invariant (LTI) system with unknown transfer function $H(z) = \frac{1}{a(z)}$.

In order to achieve this identification objective, we employ the model cepstrum, $c_h(k)$, and its relation to the poles of the model. From this relation, we find explicit solutions for the polynomial coefficients, using some well-known results on symmetric polynomials. For the white noise input signals considered here, the problem corresponds to the identification of an AR system.

The computational complexity of traditional AR system identification algorithms grows linearly with signal length. However, once the model cepstrum is known, the novel identification procedure proposed in this paper very efficiently estimates a model from only n model cepstral coefficients, with n the order of

[2] The code used to generate the results of the numerical examples can be found on https://github.com/Olauwers/Cepstral-Identification-of-Autoregressive-Systems.

[3] In this paper, we use the terms power cepstrum and cepstrum interchangeably. This name stems from the fact that it is derived from the power spectral density of a signal. There also exists a different notion, called the complex cepstrum, which we introduce in Appendix B. We always explicitly include the complex part of the name when referring to that notion.

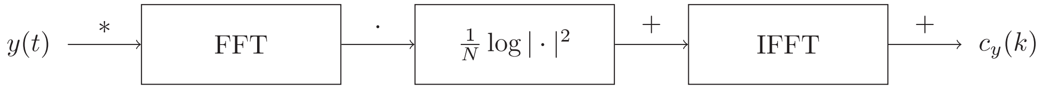


Fig. 1. The (power) cepstrum, $c_y(k)$, of a signal $y(t)$ of length N can be computed by calculating the fast Fourier transform (FFT) of the signal, squaring its magnitude, dividing the subsequent logarithm by N, and applying the inverse fast Fourier transform (IFFT) on the result. This procedure turns the convolutional structure of the time domain, through the multiplicative structure of the frequency domain, into an additive structure in the log-frequency domain. The IFFT preserves this additive structure in the (power) cepstral domain. For a more detailed discussion of the computational aspects, see Appendix C.

the model, and is, therefore, independent of the length of the cepstrum.

The proposed system identification method is related to the covariance and cepstrum matching problem (Enqvist, 2004; Ottersten, Stoica, & Roy, 1998), which aims to estimate an autoregressive moving-average (ARMA) filter from the covariances and cepstral coefficients of a signal by solving a convex optimization problem. There are, however, some important differences between the traditional approaches to this problem and the method proposed in this paper, most notably that the latter approach does not need an optimization routine, but rather provides an exact analytical solution,[4] decreasing its computational cost.

Moreover, there are instances where we have only cepstral data available (e.g., the cluster centers in the cepstral clustering problems mentioned in the introduction and discussed in detail in Section 5). Traditional covariance and cepstrum matching approaches also need the covariances, while the cepstral coefficients suffice for the proposed method (the cost of this is that the estimated model class is reduced to just AR models, rather than ARMA models). Finally, in the traditional covariance and cepstrum matching approaches, it is the MA part of the model that is estimated from the cepstral coefficients. In this paper, we provide a method that is able to identify an AR model, without needing the covariances.

Another class of system identification techniques, based on the covariance extension method (Mari et al., 2000), bear some relation to the approach developed here, and work without optimization routines. However, again, these covariance extension based methods employ the covariances of the signal, while the method in this text starts from the cepstral coefficients.

3. Symmetric polynomials

Symmetric polynomials are polynomials $P(\alpha_1,\ldots,\alpha_n)$ in n variables (i.e., the n poles α_j of the AR model), such that, for any permutation σ of the subscripts $\{1,\ldots,n\}$:

$$P\left(\alpha_{\sigma(1)},\ldots,\alpha_{\sigma(n)}\right) = P(\alpha_1,\ldots,\alpha_n)\,.$$

In this section, we exploit two special types of symmetric polynomials, the elementary symmetric and power sum polynomials, over the poles of an AR model. Well-known relations between these two types of polynomials provide us with a way to link the pole polynomial, $a(z)$, and the model cepstral coefficients, $c_h(k)$, of an AR model.

We define the *elementary symmetric polynomials*[5] as

$$e_l(\alpha_1,\ldots,\alpha_n) = \sum_{1\le j_1<\cdots<j_l\le n} \alpha_{j_1}\cdots\alpha_{j_l},$$

with $l\in\mathbb{N}$ and $e_l(\alpha_1,\ldots,\alpha_n)=0$ when $l>n$.

Vieta's theorem (Blum-Smith & Coskey, 2017) expands the pole polynomial, $a(z)$, in terms of elementary symmetric polynomials, as

$$\prod_{j=1}^{n}\left(z-\alpha_j\right) = \sum_{l=0}^{n}(-1)^l e_l(\alpha_1,\ldots,\alpha_n)\, z^{n-l}. \tag{6}$$

The coefficients a_i of the pole polynomial are then nothing more than the elementary symmetric polynomials of the poles α_j, up to their sign, or

$$a_i = (-1)^i e_i(\alpha_1,\ldots,\alpha_n)\,.$$

Furthermore, the *power sum polynomials* are defined as

$$p_l(\alpha_1,\ldots,\alpha_n) = \sum_{j=1}^{n}\alpha_j^l = l c_h(l), \quad l\in\mathbb{N}, \tag{7}$$

where the last equality follows from Eq. (5) and α_j denotes the poles of an AR model with model cepstral coefficients $c_h(l)$.

The elementary symmetric and power sum polynomials can be related to each other. These relations are known as *Newton's identities* (Boklan, 2018) and are defined recursively.

Definition 1 (*Newton's Identities*)**.** relate the elementary symmetric polynomials e_l and power sum polynomials p_l, over a set of variables $(\alpha_1,\ldots,\alpha_n)$, as

$$\begin{aligned} &l e_l(\alpha_1,\ldots,\alpha_n)\\ &\quad = \sum_{i=1}^{l}(-1)^{i-1} e_{l-i}(\alpha_1,\ldots,\alpha_n)\, p_i(\alpha_1,\ldots,\alpha_n)\,. \end{aligned}$$

Solving these recursive relations equates the elementary symmetric polynomials to combinations of power sum polynomials (dropping the variables $(\alpha_1,\ldots,\alpha_n)$ to improve readability):

$$\begin{aligned} e_1 &= p_1,\\ e_2 &= \frac{1}{2}\left(p_1^2 - p_2\right),\\ &\vdots\\ e_l &= \frac{(-1)^l}{l!}B_l(-0!p_1,\ldots,-(l-1)!p_l)\,, \end{aligned} \tag{8}$$

with $B_l(x_1,\ldots,x_l)$ the complete exponential Bell polynomials (Mihoubi, 2008), a type of polynomials that arises in combinatorics and encodes information on how a set can be partitioned.[6]

4. Cepstral identification of AR systems

Using the results above, we can very efficiently identify the pole polynomial of an AR model from its model cepstral coefficients.

[4] Traditional system identification methods using numerical linear algebra (e.g., subspace methods) "may yield a solution that is outside of the model class (e.g., non-stable models)" (Enqvist, 2004). The system identification method developed in this paper suffers from no such problems and always returns stable models (see Appendix B).

[5] For example, when $n=3$, we distinguish four different (non-zero) polynomials: $e_0(x_1,x_2,x_3)=1$, $e_1(x_1,x_2,x_3)=x_1+x_2+x_3$, $e_2(x_1,x_2,x_3)=x_1x_2+x_1x_3+x_2x_3$, and $e_3(x_1,x_2,x_3)=x_1x_2x_3$.

[6] For example, $B_3(x_1,x_2,x_3)=x_1^3+3x_1x_2+x_3$, as there is one way to partition a set of 3 variables as 3 groups of 1 element each, three ways to partition 3 variables in 1 group of 1 element and 1 group of 2 elements, and one way to partition 3 variables as 1 group of 3 elements.

Algorithm 1 Cepstral identification of AR systems

```
procedure CEPSTRAL_SYSID($c_h(k)$,$n$)
    Initialize $a(n) = 1$, with $a \in \mathbb{R}^{n\times 1}$ the vector that will contain the $n$ (= model order) coefficients of $a(z)$.
    For $l = 1 : n$ do
        $a(n-l) \leftarrow \frac{1}{l!}B_l\left(-1!c_h(1), \dots, -l!c_h(l)\right)$
    End
return $a$
```

Theorem 2. *Given an nth order stable AR model, with pole polynomial $a(z)$ and model cepstral coefficients $c_h(k)$, we have*

$$a(z) = \sum_{l=0}^{n} \frac{1}{l!} B_l\left(-1!c_h(1), \dots, -l!c_h(l)\right) z^{n-l}, \tag{9}$$

with $B_l\left(x_1, \dots, x_l\right)$ the complete exponential Bell polynomials (Mihoubi, 2008).

Proof. We start from the monic pole polynomial

$$a(z) = \prod_{i=1}^{n} \left(z - \alpha_j\right),$$

with α_j, $j \in \{1, \dots, n\}$, the poles of the AR model. Expanding in terms of elementary symmetric polynomials, as in Vieta's theorem in Eq. (6), we get

$$a(z) = \sum_{l=0}^{n} (-1)^l e_l\left(\alpha_1, \dots, \alpha_n\right) z^{n-l}.$$

Using Eq. (8), we can rewrite this expression in terms of complete exponential Bell polynomials over power sum polynomials:

$$a(z) = \sum_{l=0}^{n} \frac{1}{l!} B_l\left(-0!p_1, \dots, -(l-1)!p_l\right) z^{n-l}.$$

Finally, the relation between the power sum polynomials and the model cepstral coefficients in Eq. (7) gives us the result from the theorem,

$$a(z) = \sum_{l=0}^{n} \frac{1}{l!} B_l\left(-1!c_h(1), \dots, -l!c_h(l)\right) z^{n-l}. \quad \square$$

Using Theorem 2, we now have a system identification algorithm, as described in Algorithm 1, fulfilling the identification objective set out in Section 2.2. As the model cepstrum (Eq. (5)) falls off exponentially, it concentrates the information contained in the signal in the first few cepstral coefficients, hence we only need the first n to identify the system. If the cepstral coefficients are known, this algorithm is therefore of complexity $\mathcal{O}(n)$, with n the order of the AR model. In cases where the cepstrum is known, e.g., in cepstral clustering, which we discuss in detail in Section 5, this algorithm is thus extremely efficient, with a computational complexity that is independent of the length of the cepstrum! When the model cepstral coefficients are not known, their computation dominates the computational complexity of the identification algorithm. However, the complexity (order $\mathcal{O}(N \log N)$, with N the signal length) is still very reasonable, and will generally not overextend resources. We refer to Appendix C for the details of the computation of the cepstral coefficients.

The complete exponential Bell polynomials can be explicitly written out in terms of their arguments (Mihoubi, 2008):

$$B_n(x_1, \dots, x_n) = \sum_{k=1}^{n} \sum_{j_i} \frac{n!}{j_1! \cdots j_{n-k+1}!} \left(\frac{x_1}{1!}\right)^{j_1} \cdots \left(\frac{x_{n-k+1}}{(n-k+1)!}\right)^{j_{n-k+1}},$$

where the second summation takes place over all sequences $j_1, \dots, j_{n-k+1}$, such that

$$\begin{cases} j_1 + j_2 + \cdots + j_{n-k+1} = k \\ j_1 + 2j_2 + \cdots + (n-k+1)j_{n-k+1} = n \end{cases}.$$

With this representation, Eq. (9) is similar to the results of Schroeder (1981), who arrived at explicit expressions linking linear predictive coding (LPC) model parameters and cepstral coefficients in a different way. This builds on a body of literature in speech processing, where recursive relations for this problem have been known for quite some time. In Huang, Acero, and Hon (2001), the authors mentioned that these recursions should not be used to calculate model parameters from cepstral coefficients, however, as this might result in unstable systems. Employing the power cepstrum, rather than the complex or real cepstrum, alleviates this particular problem. The power spectral density in Eq. (3) does not take into account the phase information of the transfer function. Two systems $H(z)$ and $H(z^{-1})$ have the same power cepstrum. In the expansion of the power cepstral coefficients in terms of poles (see Eq. (5)), we always get contributions as if the poles are stable (see Appendix B), and a fortiori the resulting model is stable.

5. Cepstral clustering

We can now use this novel cepstral system identification technique to give an interpretation to time series clustering results, where we have a natural framework in which we only have access to a model cepstrum of a system that we would like to explicitly identify. We start in Section 5.1 with a short introduction to time series clustering. We show how the cepstrum can be used to define a distance measure between signals (Section 5.2) and describe a clustering approach using this cepstral distance measure (Section 5.3). Then, in Section 5.4, we interpret the center of the resulting clusters in terms of the geometric average of the power spectral densities of all signals present in these clusters.

5.1. Time series clustering

Time series clustering is the unsupervised machine learning problem of partitioning a set of time series in groups of signals that belong together in some sense. This *belonging together* is formalized in a distance measure, which quantifies how dissimilar two time series are.

As time series naturally are high-dimensional data objects, they suffer from the curse of dimensionality. The choice of distance measure is, therefore, extremely important and non-trivial, as high-dimensional spaces have some counter-intuitive geometrical properties that make defining a useful distance measure challenging. For a thorough discussion, see Verleysen and François (2005). Traditionally, shape-based distance measures, e.g., the Euclidean distance or the dynamic time warping (DTW) distance (Keogh, 2002; Lauwers & De Moor, 2017; Pereira & de Mello, 2013), are used, which more or less treat the time series as a vector, ignoring the natural correlation structure across its time dimension. Alternatively, some features are selected, and a distance measure based on these features is devised, ignoring all other information. An overview of time series clustering approaches can be found in Liao (2005).

In many engineering applications, the underlying dynamics of the time series are of interest. Hence, the shape-based or feature-based measures are seldom sufficient (Pereira & de Mello, 2013). Modeling each time series separately, on the other hand, is often impractical and computationally infeasible, especially in large-scale applications that involve several thousands or more signals. The model cepstrum, and its interpretation in model parameters in Eq. (5), allows for a model norm that can be calculated directly from the data: the *weighted cepstral distance* (see De Cock & De Moor, 2002; Lauwers & De Moor, 2017 for a more in-depth discussion). In this sense, it guarantees to quantify dissimilarity between two time series based on their generative dynamics, without the need to explicitly model these dynamics.

5.2. Weighted cepstral distance

Given two model cepstra, $c_{h_1}(k)$ and $c_{h_2}(k)$, belonging to AR models with transfer functions $H_1(z)$ and $H_2(z)$, we define the weighted cepstral distance $d\left(c_{h_1}(k), c_{h_2}(k)\right)^2$ as

$$d\left(c_{h_1}(k), c_{h_2}(k)\right)^2 = \sum_{k=1}^{\infty} k\left(c_{h_1}(k) - c_{h_2}(k)\right)^2 .$$

Denoting the simple poles of $H_1(z)$ as α_i and $H_2(z)$ as β_j, we can interpret this distance in terms of model parameters:

$$d\left(c_{h_1}(k), c_{h_2}(k)\right)^2 = \log \frac{\prod_{i=1}^{p}\prod_{j=1}^{q}\left|1-\alpha_i\bar{\beta}_j\right|^2}{\prod_{i,j=1}^{p}\left(1-\alpha_i\bar{\alpha}_j\right)\prod_{i,j=1}^{q}\left(1-\beta_i\bar{\beta}_j\right)},$$

with p and q the order of $H_1(z)$ and $H_2(z)$, respectively. We can thus compare differences between the generative dynamics of the two time series by simply calculating the weighted cepstral distance. For proofs of the above statements and a more thorough discussion, see De Cock and De Moor (2002), Lauwers and De Moor (2017).

5.3. Clustering techniques

Starting from a weighted cepstral distance matrix (i.e., a matrix, D, containing the distance between the ith and jth cepstrum at position $D[i, j]$), we can cluster the time series by using, for example, an agglomerative hierarchical method. Here, each time series initially represents a cluster of its own. These clusters are then merged iteratively, based on the shortest distance between different clusters, until the desired number of clusters is obtained. Note that the weighted cepstral distance measure is not restricted to this particular clustering technique: any clustering technique that accepts a distance matrix is viable. Clustering methods that use the Euclidean distance implicitly (e.g., k-means), could also be adapted to work on weighted versions of the cepstra. For a more in-depth description of clustering techniques, see Rokach and Maimon (2005). Cepstral clustering using hierarchical clustering was explored by some of the authors in Lauwers and De Moor (2017), but other cepstral clustering techniques can be found in Afsari, Chaudhry, Ravichandran, and Vidal (2012), Boets, De Cock, Espinoza, and De Moor (2005), Vishwanathan, Smola, and Vidal (2007) and Wolf and Shashua (2003).

5.4. Dynamical interpretation of the cluster centers

Given m cepstra, $\hat{c}_{h,j}(k)$, with $j \in \{1, \ldots, m\}$, in a cluster, and assuming that they are different realizations (i.e., coming from different input and output signals) of the same system with transfer function $H(z)$, we can calculate the average of these cepstra as

$$\tilde{c}_h(k) = \frac{1}{m}\sum_{j=1}^{m}\hat{c}_{h,j}(k). \tag{10}$$

Taking the Fourier transform of both sides, we find, from Eq. (4),

$$\begin{aligned}\mathcal{F}\left\{\tilde{c}_h(k)\right\} &= \frac{1}{m}\sum_{j=1}^{m}\log\left(\hat{\Phi}_{h,j}\left(e^{i\omega}\right)\right)\\ &= \log\left(\sqrt[m]{\prod_{j=1}^{m}\hat{\Phi}_{h,j}\left(e^{i\omega}\right)}\right)\\ &= \log\left(\tilde{\Phi}_h\left(e^{i\omega}\right)\right),\end{aligned} \tag{11}$$

where $\tilde{\Phi}_h\left(e^{i\omega}\right)$ is the geometric mean over the power spectral densities.

Given m estimates of such a power spectral density, $\hat{\Phi}_{h,j}\left(e^{i\omega}\right)$, with $j \in \{1, \ldots, m\}$, this geometric mean represents an estimator

$$\tilde{\Phi}_h\left(e^{i\omega}\right) = \sqrt[m]{\prod_{j=1}^{m}\hat{\Phi}_{h,j}\left(e^{i\omega}\right)}.$$

This estimator has been shown (Attivissimo, Savino, & Trotta, 2000; Pintelon, Schoukens, & Renneboog, 1988) to be less biased than the Euclidean average over the different $\hat{\Phi}_{h,j}\left(e^{i\omega}\right)$. The average of the cepstra $\tilde{c}_h(k)$ in Eq. (10) then represents the dynamics of the geometric mean of the different estimates of the power spectral density and characterizes the dynamics with the same low-biased estimator properties as noted for $\tilde{\Phi}_h\left(e^{i\omega}\right)$ in Pintelon et al. (1988).

Algorithm 1 allows us to estimate an AR model from the average cepstrum (i.e., cluster center) and use this as representative dynamics of the cluster. Eq. (11) states that this is equivalent to estimating dynamical models for all signals in the cluster separately and geometrically averaging them afterwards, which is computationally more complex, showing that using cepstral techniques allows us to very efficiently identify representative dynamics.

6. Numerical examples

In this section, we illustrate the theoretical results obtained in this paper. The numerical illustration in Section 6.1 serves as a proof of concept, while the numerical examples in Section 6.2, with synthetic data, investigate the properties of the novel cepstral system identification approach and show that the mean of a cepstral cluster is a good estimate of the average dynamics. We then repeat the historical identification of Yule's sunspot numbers in Section 6.3 and use the cepstrum to estimate AR models that monitor structural health in Section 6.4.

We work only on the system outputs, and thus we assume that the output cepstra correspond to the relevant model cepstra.[7] We repeat once more that all the code used during the numerical examples is available on GitHub[1].

[7] The (power) cepstrum of a white noise signal is zero everywhere except for in its 0th coefficient. In this case $c_y(k) = c_h(k)$, $\forall k \neq 0$. Therefore, we work only on the output signals (see Appendix A).

6.1. Numerical illustration

As a first practical acquaintance with cepstral system identification, we consider a simple 3rd order AR system

$$H(z) = \frac{1}{z^3 - 0.5z^2 + 0.25z - 0.125}, \tag{12}$$

which has three poles ($\alpha_1 = 0.5$, $\alpha_2 = 0.5i$, $\alpha_3 = -0.5i$).

To identify this system, we simulate the response, $y(t)$, to a Gaussian white noise input signal[6], $u(t)$, of length $N = 5 \times 10^5$. We calculate the model cepstrum, $c_h(k)$, according to the computational details in Appendix C. Algorithm 1 then yields a 3rd order AR model[8]

$$\hat{H}(z) = \frac{1}{z^3 - 0.497z^2 + 0.249z - 0.126}.$$

The poles and the transfer function of the identified model $\hat{H}(z)$ closely resemble those of the original system $H(z)$, as is also visualized in Fig. 2.

We assume in this paper that the order of the model is a user-input to the algorithm. Since Algorithm 1 is computationally very cheap once the cepstrum is known (and we only have to compute the cepstrum once, independent of the model order), in practice, we could use a simple grid search in combination with an error estimator (e.g., the Akaike information criterion Akaike, 1974) to find an appropriate value for the model order n.

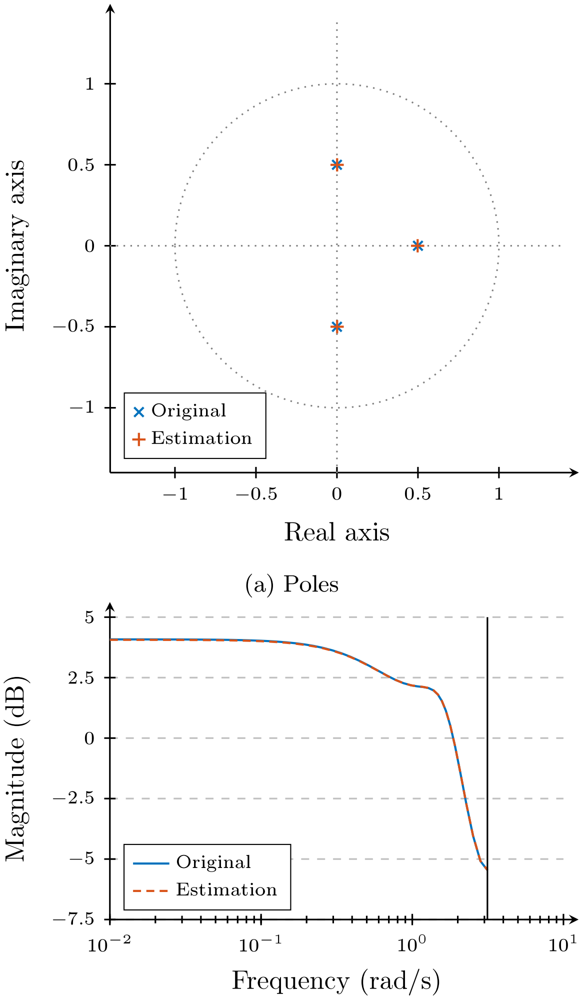
6.2. Synthetic data

We explore examples with synthetic data, where the underlying systems generating the data are known.

6.2.1. Cepstral system identification

We consider again the 3rd order AR system in Eq. (12), $H(z)$, and numerically examine the convergence behavior, noise robustness, and influence of a wrong model order. Moreover, we compare the estimation results of the cepstral system identification technique with those of the well-known ordinary least-squares regression approach, obtained by solving the normal equations directly in time domain, Burg's lattice-based method, which solves the lattice filter equations using the harmonic mean of forward and backward squared prediction errors (Brockwell & Davis, 2016; de Hoon, van der Hagen, Schoonewelle, & van Dam, 1996), and the maximum entropy method (Enqvist, 2004).

Convergence behavior. A first experiment compares the $\mathcal{H}_2$-error of the identified models for different signal lengths N. The $\mathcal{H}_2$-error is the $\mathcal{H}_2$-norm (Antoulas, 2005) of the error system $E(z) = H(z) - \hat{H}(z)$, which represents the mismatch between the original system $H(z)$ and the identified model $\hat{H}(z)$. The cepstral, least-squares, and Burg estimation approach are applied 100 times to different responses of the same signal length N of the system in Eq. (12). Furthermore, in this particular experiment, we compare the results with the maximum entropy method (Enqvist, 2004), also called the linear predictive coding (LPC) method, which interpolates a set of covariances with an AR model. Fig. 3 shows how the mean (over the 100 experiments) of the $\mathcal{H}_2$-error decreases, as expected, for longer signal lengths. The obtained results for the maximum entropy method are similar to the results obtained via the Burg estimation approach. Therefore, we do not include the maximum entropy method in the numerical results that follow in the next sections. As this convergence experiment suggests, the cepstral estimation approach seems to cope better with shorter time series than the other estimation techniques (i.e., the results for the other methods fall outside of one standard deviation of the error in this experiment, though the cepstral estimation lies inside one standard deviation of the error of the other methods; further research on this topic is required).



(a) Poles

(b) Transfer functions

Fig. 2. The model identified via the cepstral system identification algorithm closely resembles the original system $H(z)$. The original poles are $\alpha_1 = 0.5$, $\alpha_2 = 0.5i$, and $\alpha_3 = -0.5i$.

Noise robustness. In Fig. 4, we investigate the influence of noise on the identification results. The response $y(t)$ simulated by the system in Eq. (12) is contaminated with additive white Gaussian noise of different magnitudes σ: $\tilde{y}(t) = y(t) + \sigma n(t)$. For high signal-to-noise ratios (SNRs), the identified transfer functions behave almost exactly as the original one. Down to 10 dB, the results remain quite satisfactory. When the SNR drops below 10 dB, the model estimated by the cepstral identification procedure deviates significantly from the original. The least-squares and Burg estimation approach yield similar results.

Wrong model order. Now, we add a conjugate pair of poles to the 3rd order AR system in Eq. (12), $H(z)$, to obtain a 5th order AR system

$$G(z) = \frac{H(z)}{z^2 + 0.5z + 0.125} = \frac{1}{(z - 0.5)(z^2 + 0.25)(z^2 + 0.5z + 0.125)}.$$

[8] The cepstral coefficients and exact model parameters depend on the exact noise realization (i.e., the random seed).

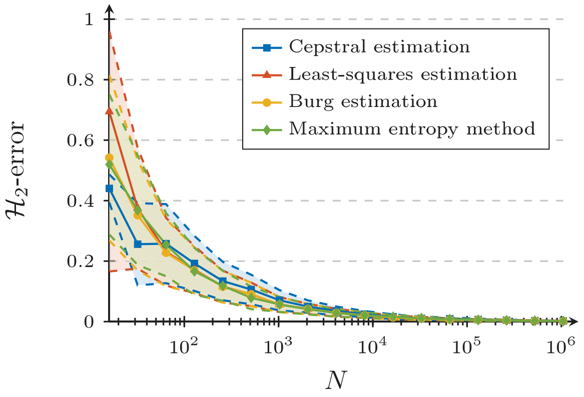


Fig. 3. The cepstral, least-squares, Burg, and maximum entropy estimation approach are applied 100 times to different output signals of the same length N coming from $H(z)$ in Eq. (12). The solid line shows how the mean of the $\mathcal{H}_2$-error decreases for longer data lengths, while the dashed lines represent the standard deviation. The cepstral system identification procedure seems to cope better with shorter time series than the other estimation techniques (i.e., the results for the other methods fall outside of one standard deviation of the error in this experiment, though the cepstral estimation lies inside one standard deviation of the error of the other methods; further research on this topic is required).

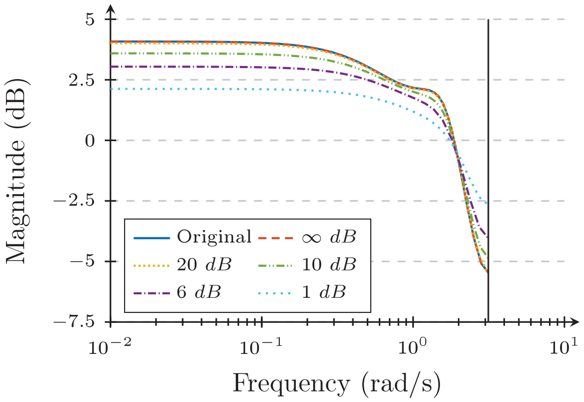


Fig. 4. A visualization of the influence of the signal-to-noise ratio (SNR) on the identification results. For high SNRs, the identified transfer functions behave almost exactly as the original one. Down to 10 dB, the results remain quite satisfactory. However, when the SNR drops below 10 dB, the model estimated by the cepstral identification procedure deviates significantly from the original system. The least-squares and Burg estimation approach yield similar results.

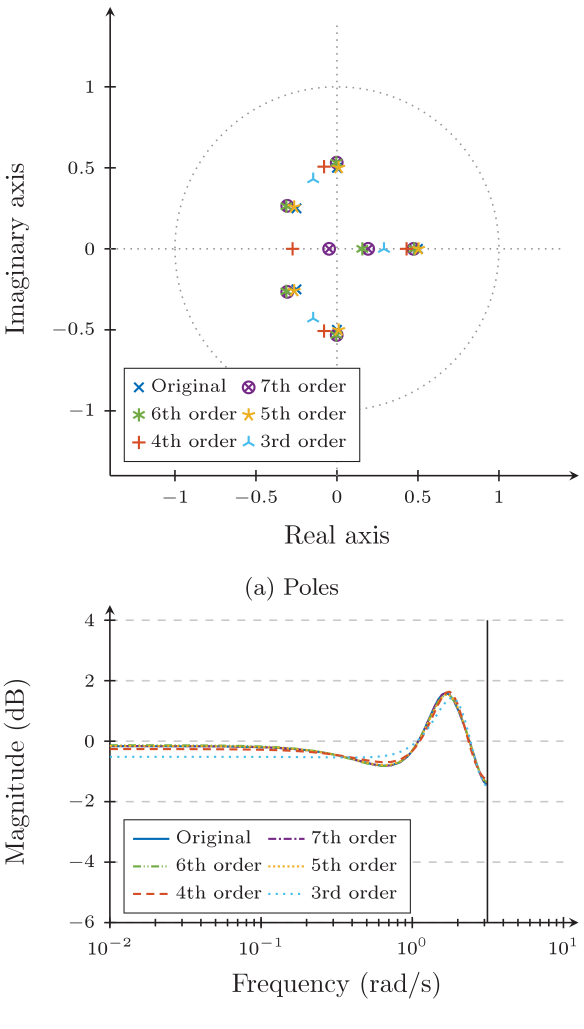
When we try to identify this 5th order system with models of different order, we notice a clear decrease in the identification performance, which is of course to be expected. Fig. 5 compares the resulting poles and transfer functions for a 7th, 6th, 5th, 4th, and 3rd order model. The correct model order ($n = 5$) clearly results in decent identification results, while lower model orders ($n = 4$ and $n = 3$) fail to capture all the system dynamics. Higher order models ($n = 7$ and $n = 6$) result visually in close magnitude plots, although they do give similar $\mathcal{H}_2$-errors as the lower order models in Fig. 6. Fig. 6 also demonstrates that a longer signal length N does not alleviate the model mismatch problem.

6.2.2. Mean of a cepstral cluster

We now turn our attention to the interpretation of the average cepstrum in Eq. (11), in the context of cepstral clustering.

Consider two systems,

$$H_1(z) = H(z) = \frac{1}{z^3 - 0.5z^2 + 0.25z - 0.125},$$



(a) Poles

(b) Transfer functions

Fig. 5. The 5th order model identified via the cepstral estimation procedure closely resembles the original system $G(z)$, while the other models deviate from the original system. The original poles are $\alpha_1 = 0.5$, $\alpha_2 = 0.5i$, $\alpha_3 = -0.5i$, $\alpha_4 = -0.25 + 0.25i$, and $\alpha_5 = -0.25 - 0.25i$.

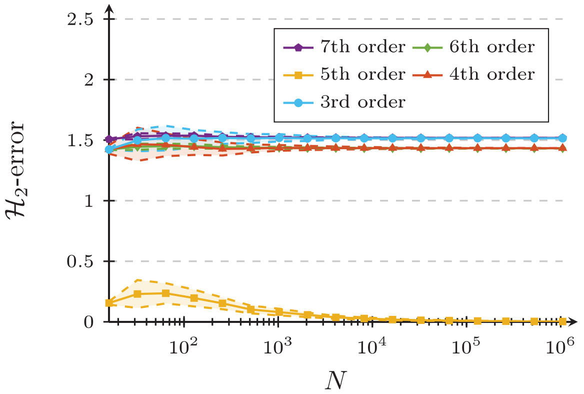


Fig. 6. The cepstral estimation approach for different model orders is applied 100 times to data coming from the original system $G(z)$ with different (same-length) realizations of the white Gaussian input. The solid line shows the mean of the $\mathcal{H}_2$-error for the 5th order estimation, decreasing with longer data lengths. However, when the model order is different from 5, longer signals do not improve the $\mathcal{H}_2$-error.

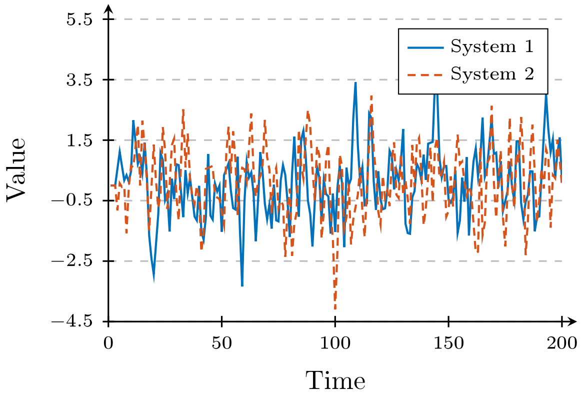


Fig. 7. Two output signals, which are realizations of systems $H_1(z)$ in Eq. (12) and $H_2(z)$ in Eq. (13), respectively, with Gaussian white noise as input.

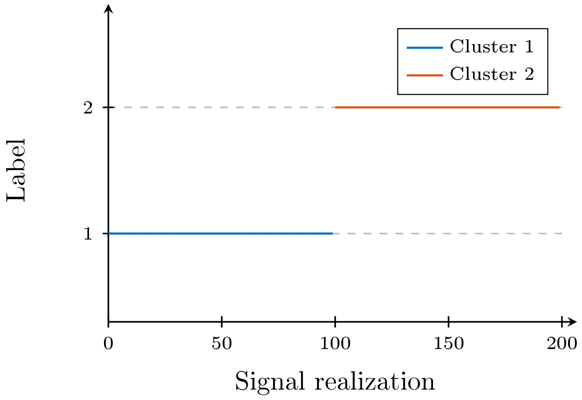


Fig. 8. Results of agglomerative hierarchical clustering, using the weighted cepstral distance. The first 100 signals (0–99) are outputs of $H_1(z)$ in Eq. (12), the last 100 belong to system $H_2(z)$ in Eq. (13). Cepstral clustering separates them perfectly in two clusters, each representing one of the two systems. Examples of signals belonging to both clusters are shown in Fig. 7.

as in Eq. (12), and

$$H_2(z) = \frac{1}{z^3 - 0.4z^2 + 0.3025z - 0.121}, \tag{13}$$

with poles $\alpha_1 = 0.4$, $\alpha_2 = 0.55i$, and $\alpha_3 = -0.55i$.

We simulate each system 100 times with Gaussian white noise input signals and denote the output signals of $H_1(z)$ as $y_{1,i}(t)$, for $i \in \{1, \dots, 100\}$, with cepstra $c_{1,i}(k)$, and those of $H_2(z)$ as $y_{2,j}(t)$, for $j \in \{1, \dots, 100\}$, with cepstra $c_{2,j}(k)$. Two such output signals, one for each system, are shown in Fig. 7. The poles of these two systems are purposely chosen close together, so that the output signals are similar and the clustering challenge is non-trivial. Performing cepstral agglomerative hierarchical clustering (see Section 5.3) on this data set, cutting off at two clusters, exactly separates the signals coming from each system. The first cluster, C_1, contains all cepstra $c_{1,i}(k)$, while the second cluster, C_2, contains all cepstra $c_{2,j}(k)$. Fig. 8 shows the resulting labels.

Given a model cepstrum, we can identify, via Algorithm 1, the corresponding pole polynomial and obtain the power spectral density (PSD) of the identified model very efficiently. Fig. 9 shows PSDs obtained in this way for various elements of cluster C_1, to show the variance on the dynamics present in the data set. In Fig. 10, we show, for cluster C_1, the true PSD (i.e., the PSD associated with the system $H_1(z)$), the geometric mean of all PSDs (i.e., first identify the PSD for each signal in C_1, by using the system identification technique from this paper to estimate the pole polynomial, and then compute the frequency response; the geometric average over all PSDs obtained in this way is shown), and the PSD associated with the cluster center (i.e., the system identification technique applied to the cluster center, which itself is the average over all the cepstra present in the cluster). We compare this with the PSD of the Euclidean average over all the signals in the cluster. It is clear that the cluster center provides a good estimate for the average dynamics of the cluster, and certainly a better one than the Euclidean average. We can thus interpret the cluster centers obtained by cepstral clustering as good estimates of the average dynamics present in their clusters. Instead of having to find a model for each realization of the system and taking the geometric mean, it suffices to model the cluster center.

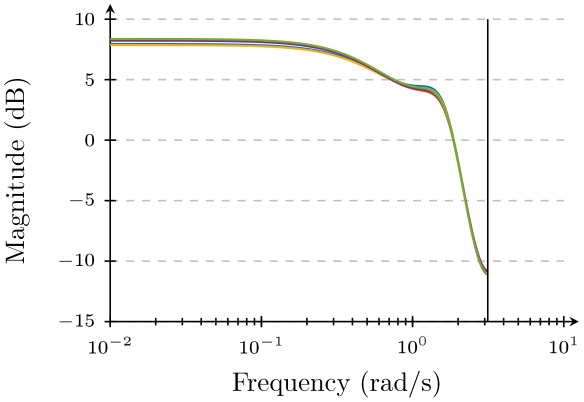


Fig. 9. Power spectral density (PSD) estimates of some of the signals in $y_{1,i}(t)$, the output signals of system $H_1(z)$ in Eq. (12). The PSD estimates clearly exhibit some variance.

6.3. Yule's sunspot numbers

Already in 1927, Yule (Yule, 1927) proposed the structure that we know today as an AR model. He introduced this statistical technique to describe the historical series of recorded yearly sunspot numbers, a time series of visual sunspot counts initiated in 1848 by the Swiss astronomer Johann Rudolf Wolf (Werner, 2012). Yule, however, called this series Wolfer's sunspot numbers, in honor of Wolf's successor at the Zürich observatory, Alfred Wolfer. These sunspot numbers are an important indicator for solar activity and the only direct information available to retrace the long-term evolution of the solar cycle, but also play a role in more recent discussions, like climate change, space flights, or telecommunication (Clette, Svalgaard, Vaquero, & Cliver, 2014; Werner, 2012).

Although the quality of the original data is questionable (Werner, 2012), especially the oldest observations, we repeat this historical identification experiment on the same data as Yule, including the initial detrending[9] (see Fig. 11), with yearly sunspot numbers between 1749 and 1924, resulting in a time series of 176 sunspot numbers.

As expected, the cepstral, least-squares, and Burg approach (for $n = 2$) closely match Yule's historical model after the initial detrending (see numerical results in Table 1). Fig. 12 compares the results visually. Moreover, the cepstrum seems to cope very well with the trend in the original data. Fig. 13 shows how the identification results change when we do not detrend the time series. The least-squares and Burg approach completely fail, while the cepstral estimation results in a more or less satisfactory model that captures most of the system dynamics. This

[9] The detrending removes the best straight-line fit linear trend from the data.

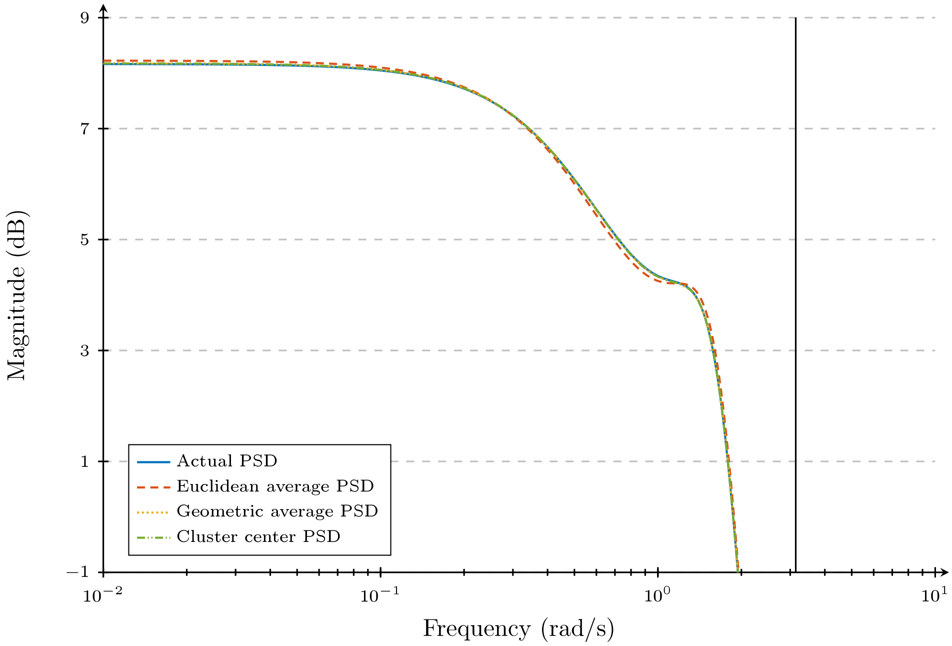


Fig. 10. Power spectral density (PSD) estimates for various quantities, compared to the actual PSD. All of these results are for the output signals of system $H_1(z)$, in Eq. (12). The geometric average PSD is the geometric average over the PSDs belonging to all the output signals of system $H_1(z)$ (i.e., we identified a model for each signal, computed all their PSDs, and took the geometric average). The cluster center PSD is the one belonging to the cepstral cluster center. Note that this PSD is equal to the geometric average, illustrating Eq. (11), and they are both good estimators of the actual PSD. The Euclidean average PSD is the PSD belonging to the Euclidean average over all signals in cluster C_1. It deviates from the actual PSD, showing the merit of cepstral clustering over traditional approaches for the purpose of finding the average dynamics.

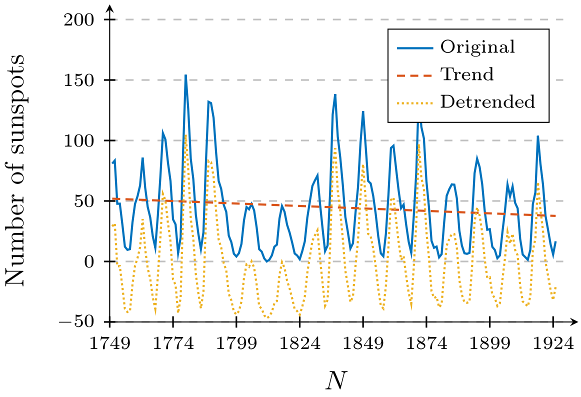


Fig. 11. This figure visualizes Yule's sunspot numbers before and after detrending.

Table 1
Numerical results of the identification of Yule's historical sunspot data. The pole polynomials $a(z) = z^2 + a_1 z + a_2$ are obtained via cepstral, least-squares, and Burg estimation on the detrended and trended (indicated with *) time series.

Approach	a_1	a_2
Yule's historical model	−1.3425	0.6550
Cepstral estimation	−1.3317	0.6587
Cepstral estimation *	−1.4678	0.7412
Least-squares estimation	−1.3345	0.6517
Least-squares estimation *	−1.4319	0.5444
Burg estimation	−1.3368	0.6516
Burg estimation *	−1.4381	0.5475

stems from the fact that the cepstrum stores information on the scale of the problem in its 0th coefficient, which is not used in the identification technique. Part of the detrending is thus done automatically.

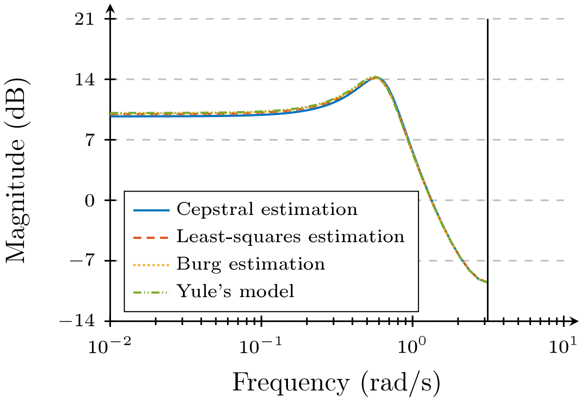


Fig. 12. The identification of Yule's historic sunspot numbers via the cepstral, least-squares, and Burg approach. All methods yield a close match to Yule's model.

6.4. Structural health monitoring via AR models

Many civil and mechanical systems require close structural health monitoring (SHM) systems, both from an economic and a safety perspective (Sohn, Czarnecki, & Farrar, 2000). Damage identification based on changes in dynamic response to vibrations is one of the few methods to monitor structural health in a non-destructive way. One approach is the use of AR models to identify damage in given structures. The residuals (Bornn, Farrar, Park, & Farinholt, 2009; Fugate, Sohn, & Farrar, 2001) and the coefficients (Sohn et al., 2000) of the AR models fitted on motion measurements of vibrating structures serve as important features to detect structural damage.

We apply this AR approach on the measured response of a vibrating three-story frame structure (sketched in Fig. 14) and use the coefficients of the fitted models to identify structural damage.

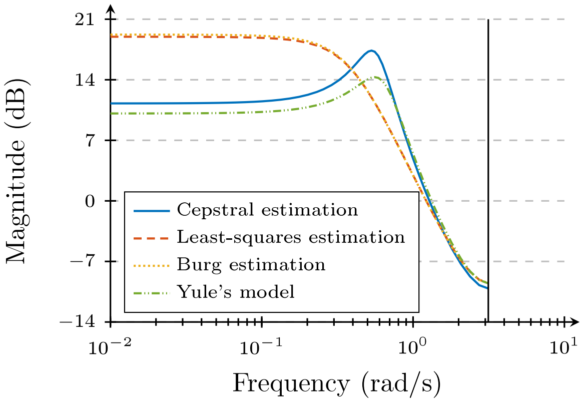


Fig. 13. When we do not detrend the sunspot number data before identification, the least-squares and Burg approach fail completely to identify the system. However, the cepstral estimation results in a rather satisfactory result that captures most of the system dynamics.

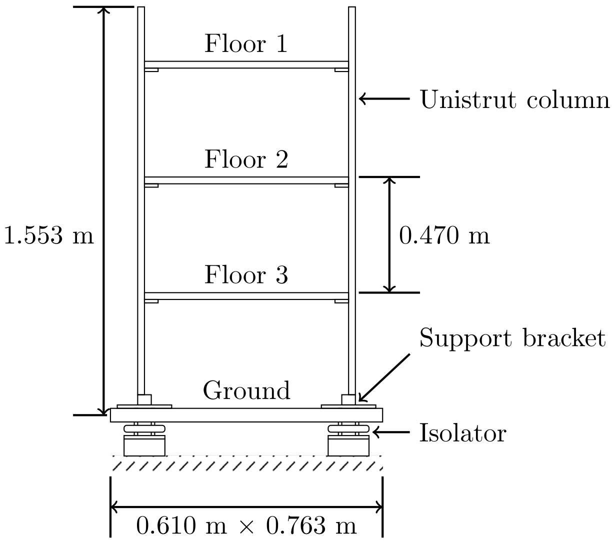


Fig. 14. The three-story test setup of which we monitor the structural health consists of an aluminum frame and four aluminum floor plates, connected through bolted connections. In undamaged state, all bolted connections are tightened properly. However, to simulate structural damage, we loosen or remove some bolts. At its bottom, a shaker vibrates the structure and several accelerometers measure the resulting movements of the different aluminum parts.

The structure consists of an aluminum frame with floor plates, connected through bolted connections. In undamaged state, all bolts are tightened properly. By loosening or removing some bolts, structural damage is simulated. The frame is excited by a shaker at its bottom. Several accelerometers measure the movements of the different aluminum parts.[10]

We follow the approach as described in the paper of Sohn et al. (2000), where similar experiments are done on a bridge structure. We identify the AR coefficients (order $n = 3$) of non-overlapping windows (window length $M = 50$) of the output of an accelerometer close to the defect, using the cepstral identification approach in Algorithm 1. The interested reader finds information about the influence of these design parameters on the structural damage detection in Figueiredo, Figueiras, Park, Farrar, and Worden (2011).

Input data on the movement of the shaker was not available. We thus, again, have to make the assumption that the dynamics contained in the output signal constitute the dynamics of the relevant model. In the cepstral domain, this translates to assuming that $c_y(k) = c_h(k)$. Results are shown in Fig. 15. The experiment was done twice without damage, for different levels of excitation. The first experiment drives 2 volts through the shaker (undamaged 2V), the second one applies 5 volts (undamaged 5V). We notice that the AR coefficients remain within the same range, for different levels of excitation of the shaker. We then repeat the experiment with structural damage (removal of the bolt named L1C). The damaged AR coefficients in Fig. 15 clearly differ from those in the undamaged situation, especially for the second and third coefficient (the first coefficient seems to be less sensitive to damage, an observation that has also been made by Sohn et al., 2000). These differences in AR coefficients allow us to distinguish undamaged and damaged structures.

Finally, we replicate the results with the least-squares and Burg estimation approach. However, the differences between the damaged and undamaged AR coefficients are not as distinctive (see for example the third coefficient in Fig. 16). Perhaps the rather short window length is to blame here, in accordance with the results in Fig. 3.

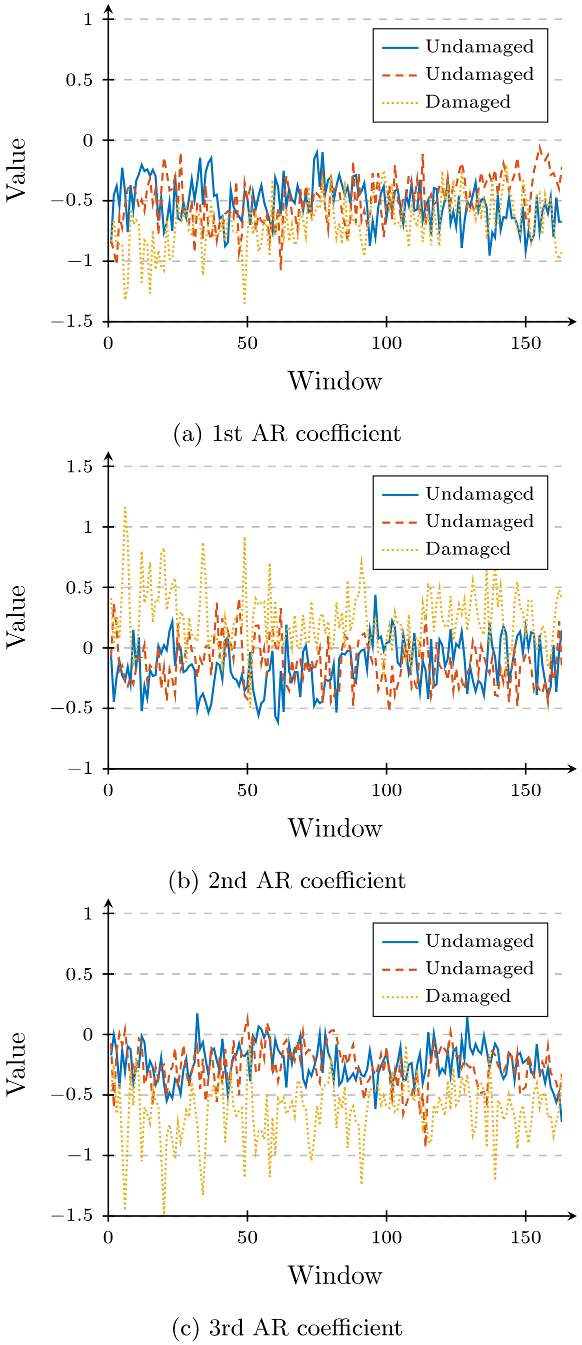
7. Conclusion and future research

In this paper, we developed a novel general framework to identify the pole polynomial, $a(z)$, of an unknown autoregressive (AR) system, exploiting the link between the model power cepstrum and the poles of the model, which resulted in analytic solutions for the coefficients of $a(z)$. The computation of the power cepstrum is the main contributor to the computational complexity of the system identification procedure presented in Algorithm 1. This complexity is $\mathcal{O}(N \log N)$, with N the signal length, resulting in a fast system identification technique that can be applied to large-scale problems. When the power cepstrum is known, e.g., in cepstral clustering problems, the procedure is independent of the length of the power cepstrum and is of complexity $\mathcal{O}(n)$, with n the model order.

Via numerical examples with synthetic data, we investigated the convergence speed and robustness (to measurement noise and wrong model order selection) of the identification procedure. The algorithm was applied to a clustering problem, to show how the cepstral cluster center provides a good estimator for the average dynamics of the cluster (and therefore of the generative model of the cluster). The classical example of Yule's sunspot numbers and a contemporary challenge in structural health monitoring corroborated the theoretical results. We also compared our novel approach to the traditional least-squares and Burg estimation approach.

In future work, we will explore several extensions of the current framework, to include models with zeros as well as poles (the zeros would enter with a minus sign in the power sums in Eq. (7), which offers a way to separate the contributions of poles and zeros) and allow for multiple-input multiple-output models (using the MIMO extension of the power cepstrum from Lauwers, Agudelo, & De Moor, 2018). We will pursue ways to employ the information in higher cepstral coefficients to improve parameter estimation, model order selection, and robustness. We will also further investigate theoretically the differences with the traditional least squares and Burg approach, to understand the differences in performance, for example for shorter signals.

[10] More information about the test setup or the data set can be found on the website of Los Alamos National Laboratory (https://www.lanl.gov/projects/national-security-education-center/engineering/software/shm-data-sets-and-software.php). The data set used in this example is "Bookshelf Frame Structure – DSS 2000".



(a) 1st AR coefficient

(b) 2nd AR coefficient

(c) 3rd AR coefficient

Fig. 15. The AR coefficients of the 3rd order AR model fitted on the output of an accelerometer clearly indicate whether the frame contains some structural damage. Results of two experiments without and one experiment with structural damage are shown. Especially the second and third damaged AR coefficients differ from their undamaged equivalent.

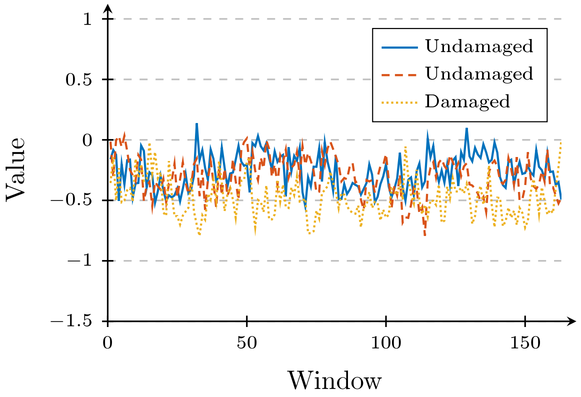


Fig. 16. The 3rd AR coefficient for each window, now obtained with Burg estimation. Results for the least-squares approach are similar, and therefore omitted. The differences of the AR coefficients are less distinctive for these two approaches. We suspect that the length of the non-overlapping windows is the main culprit of this behavior, as this problem seems to disappear with longer window sizes.

Acknowledgments

This work was supported in part by the KU Leuven Research Fund, Belgium (projects C16/15/059, C3/19/053, C32/16/013, C24/18/022), in part by the Industrial Research Fund, Belgium (Fellowship 13-0260) and several Leuven Research and Development bilateral industrial projects, Belgium, in part by Flemish Government Agencies: FWO, Belgium (EOS project no. 30468160 (SeLMA), SBO project S005319N, infrastructure project I013218N, TBM project T001919N, and PhD grants (SB/1SA1319N, SB/1S93918, SB/151622)), Flanders AI Research Program, Belgium, and VLAIO, Belgium (City of Things (COT.2018.018), industrial project HBC.2018.0405, and PhD grants (Baekeland mandate HBC.20192204 and innovation mandate HBC.2019.2209)), in part by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (grant agreement 885682), and in part by the European Commission (EU H2020-SC1-2016-2017 grant agreement 727721: MIDAS and ERC grant). Oliver Lauwers and Christof Vermeersch are both supported by FWO SB Research Fellowships. Oliver Lauwers is also supported by the KIC InnoEnergy and Cepsis.

Appendix A. General inputs

All results in this paper hold for stationary inputs. However, Theorem 2 is written in terms of the model (power) cepstrum. As long as we have access to these cepstral coefficients, we can use the theorem. In this appendix, we describe how to obtain the model cepstrum in the case of non-white noise inputs (i.e., $c_h(k) \neq c_y(k)$).

This notion of the cepstrum stems from the field of *homomorphic signal processing* (Oppenheim & Schafer, 1975). Eq. (4) turns the convolutional structure of the time domain, through the Fourier transform and the logarithm, into an additive structure. The inverse Fourier transform then turns this back into a transformed version of the time domain. This is known as *quefrency alanysis* and has given rise to a lot of similar terms, such as liftering instead of filtering and rhamonics instead of harmonics (Bogert, 1963).

Formally, in time domain, taking the inverse Fourier transform of the transfer function returns the impulse response, $h(t)$, and the following relation holds:

$$y(t) = h(t) * u(t),$$

with $y(t)$ and $u(t)$ the time domain output and input signals, respectively, and $*$ the convolution operator. In light of this, it is easy to see that

$$c_h(k) = c_y(k) - c_u(k). \tag{A.1}$$

This model cepstrum can be interpreted as the cepstrum of the impulse response, $h(t)$. It can always be computed when the Fourier transforms of the signals are well-defined.

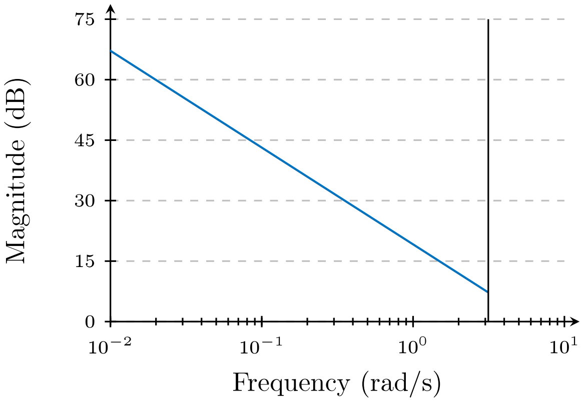


Fig. A.1. The power spectral density (PSD) of the colored (non-white) noise used in Fig. A.2.

For example, when we simulate the system in Eq. (12), $H(z)$, with a non-white noise input signal (e.g., the colored noise in Fig. A.1), the model cepstral coefficients, $c_h(k)$, no longer correspond to the output cepstral coefficients, $c_y(k)$, and an identification using only the output fails to capture the system dynamics. However, if we know the input cepstrum, $c_u(k)$, Eq. (A.1) gives us the correct model cepstral coefficients and allows us to properly estimate the AR model (see Fig. A.2).

Appendix B. On the assumption of stability

Because of the use of the power spectral density in Eq. (4), the power cepstrum does not take into account any information about stability of the systems (i.e., whether poles have magnitudes greater or smaller than 1). For any unstable pole γ (i.e., $|\gamma| > 1$), the contribution to the model power cepstral coefficient, $c_h(k)$, $\forall k \neq 0$, will be $\frac{\gamma^{-k}}{k}$ (Oppenheim & Schafer, 1975). If we would then perform the system identification procedure from this paper, we will identify a pole γ^{-1}, and always obtain a stable system.

If we want to allow unstable systems, we can instead employ the *complex cepstrum*,[11] defined as

$$\hat{c}_h(k) = \mathcal{F}^{-1}\left\{\log\left(H\left(e^{i\omega}\right)\right)\right\},$$

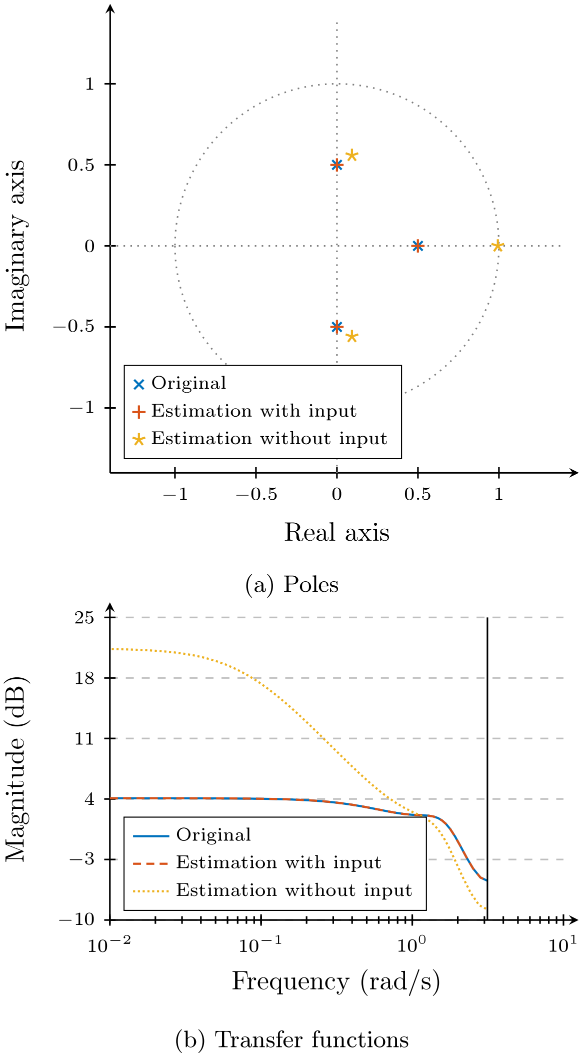
and computed as

$$\hat{c}_y(k) = \text{IFFT}\left(\log\left(\text{FFT}(y(t))\right)\right).$$

Consider now a system

$$Y(z) = \frac{1}{b(z)c(z)}U(z), \tag{B.1}$$

where $b(z)$ contains the stable poles, $|\beta_i| < 1$, and $c(z)$ the unstable poles, $\left|\gamma_j\right| > 1$. The question that we want to answer now is whether we can, based on input–output signal pairs only, determine whether the system is stable or not and identify its pole polynomial $a(z) = b(z)c(z)$.



(a) Poles

(b) Transfer functions

Fig. A.2. The model estimated via the cepstral system identification procedure closely resembles the original system $H(z)$ when the colored (non-white) noise of Fig. A.1 is taken into account. However, when the non-white input is not taken into account, the identification clearly misestimates the original poles ($z_1 = 0.5$, $z_2 = 0.5i$, and $z_3 = -0.5i$).

We can calculate its model complex cepstrum, $\hat{c}_h(k) = \hat{c}_y(k) - \hat{c}_u(k)$, and it is easily shown that

$$\hat{c}_h(k) = \begin{cases} \displaystyle\sum_{i=1}^{q} \frac{\beta_i^k}{k} & \forall k > 0 \\ \log(g') & k = 0 \\ \displaystyle -\sum_{j=1}^{s} \frac{\gamma_j^k}{k} & \forall k < 0. \end{cases}$$

In other words, the complex cepstrum stores the cepstral coefficients belonging to the stable poles in positive values of k and the coefficients belonging to the unstable poles in negative values of k. We show an example of the complex cepstrum in Fig. B.1. We can run Algorithm 1 on the positive and negative sequences separately. This gives us the pole polynomials $b(z)$ and $c(z^{-1})$ in Eq. (B.1). We then only have to substitute $z \to z^{-1}$ for the unstable part, multiply the two polynomials, to find

$$a(z) = b(z)c(z).$$

[11] The term complex cepstrum is a bit confusing, as the complex cepstral coefficients are real numbers. Rather, the phase information of the transfer function is retained in computing the complex cepstrum, giving rise to its name.

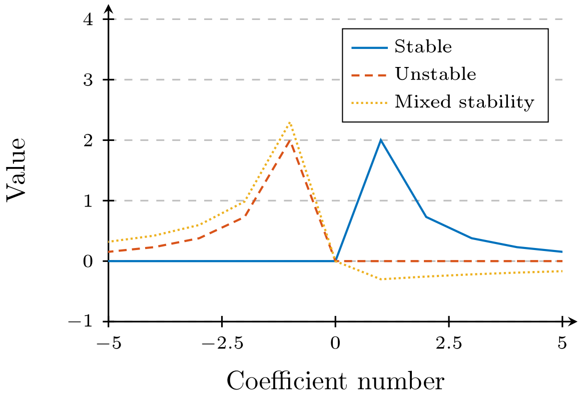


Fig. B.1. The first five positive and negative complex cepstral coefficients of a stable system (poles at 0.9, 0.7, 0.4), an unstable system (poles at 1/0.9, 1/0.7, 1/0.4) and a mixed system (poles at 1/0.9, 0.7, 1/0.4), respectively. The 0th cepstral coefficient has been artificially set to 0 in all three cases to improve readability of the graph. This coefficient can take on large (positive or negative) values, but is not important for the discussion here.

Appendix C. Computational aspects

The (power) cepstrum of a signal $y(t)$ can be straightforwardly computed as

$$c_y(k) = \text{IFFT}\left(\log\left(\Phi_y\left(e^{i\omega}\right)\right)\right),$$

with $k = 1, \ldots, N$, and N the length of the fast Fourier transform (FFT). Both the implementation of the inverse fast Fourier transform (IFFT) (Brigham, 1988) and the logarithm are straightforward in this case and are pre-implemented in many commonly used scientific programming languages, like MATLAB and Python. Computing a good estimate of the power spectral density (PSD) $\Phi_y\left(e^{i\omega}\right)$ is a little more involved. For long enough signals (from about $N = 2^{10}$ and beyond), we can employ the well-known FFT as an approximation of the Fourier transform and implement

$$\Phi_y\left(e^{i\omega}\right) = \frac{1}{N}\left|\text{FFT}(y(t))\right|^2.$$

The FFT has a computational complexity of $\mathcal{O}(N \log N)$.

The accuracy of the PSD estimate is very important to our techniques. For shorter time series, we employ *Welch's method* (Welch, 1967), which divides the signal in overlapping windows, estimates the PSD of each of the windows using the FFT, and averages them out. The result is a less noisy estimate. This technique is again of $\mathcal{O}(N \log N)$, and is what we use throughout this paper (see Algorithm 2). For shorter time series (i.e., $N < 2^7$), even more accurate estimation techniques exist (e.g., the *Multitaper method* Percival & Walden, 1993), but these are computationally more expensive and are, therefore, to be avoided for longer signals.

Algorithm 2 Power cepstrum computation

```
procedure POWER_CEPSTRUM(y(t))
    $\Phi_y\left(e^{i\omega}\right) \leftarrow \text{Welch}(y(t))$
    $\Phi'_y\left(e^{i\omega}\right) \leftarrow \log\left(\Phi_y\left(e^{i\omega}\right)\right)$
    $c_y(k) \leftarrow \text{IFFT}\left(\Phi'_y\left(e^{i\omega}\right)\right)$
return $c_y(k)$
```

References

Afsari, Bijan, Chaudhry, Rizwan, Ravichandran, Avinash, & Vidal, René (2012). Group action induced distances for averaging and clustering linear dynamical systems with applications to the analysis of dynamic scenes. In *Proc. of the 2012 IEEE conference on computer vision and pattern recognition* (pp. 2208–2215).

Akaike, Hirotugu (1974). Stochastic theory of minimal realization. *IEEE Transactions on Automatic Control*, *19*(6), 667–674.

Antoulas, Athanasios C. (2005). *Advances in design and control*, *Approximation of large-scale dynamical systems*. Philadelphia, PA, USA: SIAM.

Attivissimo, Filippo, Savino, Mario, & Trotta, Amerigo (2000). A study on nonlinear averagings to perform the characterization of power spectral density estimation algorithms. *IEEE Transactions on Instrumentation and Measurement*, *49*(5), 1036–1042.

Blum-Smith, Ben, & Coskey, Samuel (2017). The fundamental theorem on symmetric polynomials: History's first whiff of Galois theory. *The College Mathematics Journal*, *48*(1), 18–29.

Boets, Jeroen, De Cock, Katrien, Espinoza, Marcelo, & De Moor, Bart (2005). Clustering time series, subspace identification and cepstral distances. *Communications in Information & Systems*, *5*(1), 69–96.

Bogert, Bruce P. (1963). The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. In *Proc. of the symposium on time series analysis, vol. 15* (pp. 209–243).

Boklan, Kent D. (2018). A note on identities for elementary symmetric and power sum polynomials. *Discrete Mathematics, Algorithms and Applications*, *10*(1).

Bornn, Luke, Farrar, Charles R., Park, Gyuhae, & Farinholt, Kevin (2009). Structural health monitoring with autoregressive support vector machines. *Journal of Vibration and Acoustics*, *131*(2), 1–9.

Brigham, Oran E. (1988). *The fast fourier transform and its applications*. Upper Saddle River, NJ, USA: Prentice-Hall.

Brockwell, Peter J., & Davis, Richard A. (2016). *Springer texts in statistics*, *Introduction to time series and forecasting* (3rd ed.). Cham, Switzerland: Springer.

Clette, Frédéric, Svalgaard, Leif, Vaquero, José M., & Cliver, Edward W. (2014). Revisiting the sunspot number. *Space Science Reviews*, *186*(1–4), 35–103.

De Cock, Katrien, & De Moor, Bart (2002). Subspace angles between ARMA models. *Systems & Control Letters*, *46*(4), 265–270.

de Hoon, Michiel J. L., van der Hagen, Tim H. J. J., Schoonewelle, Hielke, & van Dam, Hugo (1996). Why Yule–Walker should not be used for autoregressive modelling. *Annals of Nuclear Energy*, *23*(15), 1219–1228.

Edwards, John (2019). Signal processing improves autonomous vehicle navigation accuracy: Guidance innovations promise safer and more reliable autonomous vehicle operation. *IEEE Signal Processing Magazine*, *36*(2), 15–18.

Enqvist, Per (2004). A convex optimization approach to ARMA(n,m) model design from covariance and cepstral data. *SIAM Journal on Control and Optimization*, *43*(3), 1011–1036.

Figueiredo, Eloi, Figueiras, Joaquim, Park, Gyuhae, Farrar, Charles R., & Worden, Keith (2011). Influence of the autoregressive model order on damage detection. *Computer-Aided Civil and Infrastructure Engineering*, *26*(3), 225–238.

Fugate, Michael L., Sohn, Hoon, & Farrar, Charles R. (2001). Vibration-based damage detection using statistical process control. *Mechanical Systems and Signal Processing*, *15*(4), 707–721.

Huang, Xuedong, Acero, Alex, & Hon, Hsiao-Wuen (2001). *Spoken language processing: a guide to theory, algorithm, and system development*. Upper Saddle River, NJ, USA: Prentice-Hall.

Keogh, Eamonn (2002). Exact indexing of dynamic time warping. In *Proc. of the 28th international conference on very large data bases* (pp. 406–417).

Kim, Jayoung, Campbell, Alan S., Esteban-Fernández de Ávila, Berta, & Wang, Joseph (2019). Wearable biosensors for healthcare monitoring. *Nature biotechnology*, *37*(4), 389–406.

Lauwers, Oliver, Agudelo, Oscar Mauricio, & De Moor, Bart (2018). A multiple-input multiple-output cepstrum. *IEEE Control Systems Letters*, *2*(2), 272–277.

Lauwers, Oliver, & De Moor, Bart (2017). A time series distance measure for efficient clustering of input/output signals by their underlying dynamics. *IEEE Control Systems Letters*, *1*(2), 286–291.

Liao, Warren T. (2005). Clustering of time series data—A survey. *Pattern Recognition*, *38*(11), 1857–1874.

Ljung, Lennart (2010). Perspectives on system identification. *Annual Reviews in Control*, *34*(1), 1–12.

Mari, Jorge, Dahlén, Anders, & Lindquist, Anders (2000). A covariance extension approach to identification of time series. *Automatica*, *36*(3), 379–398.

Mihoubi, Miloud (2008). Bell polynomials and binomial type sequences. *Discrete Mathematics*, *308*(12), 2450–2459.

Oppenheim, Alan V., & Schafer, Ronald W. (1975). *Prentice-hall international editions, Digital signal processing* (1st ed.). Englewood Cliffs, NJ, USA: Prentice-Hall.

Ottersten, Björn, Stoica, Peter, & Roy, Richard (1998). Covariance matching estimation techniques for array signal processing applications. *Digital Signal Processing*, *8*(3), 185–210.

Percival, Donald B., & Walden, Andrew T. (1993). *Spectral analysis for physical applications*. Cambridge, UK: Cambridge University Press.

Pereira, Cássio Martini Martins, & de Mello, Rodrigo F. (2013). Common dissimilarity measures are inappropriate for time series clustering. *Revista de InformÁTica TeÓRica E Aplicada*, *20*(1), 25–48.

Perera, Charith, Zaslavsky, Arkady, Christen, Peter, & Georgakopoulos, Dimitrios (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, *16*(1), 414–454.

Pintelon, Rik, Schoukens, Johan, & Renneboog, Jean (1988). The geometric mean of power (Amplitude) spectra has a much smaller bias than the classical arithmetic (RMS) averaging. *IEEE Transactions on Instrumentation and Measurement*, *37*(2), 213–218.

Rokach, Lior, & Maimon, Oded (2005). Clustering methods. In Oded Maimon, & Lior Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 321–352). Boston, MA, USA: Springer.

Schroeder, Manfred R. (1981). Direct (nonrecursive) relations between cepstrum and predictor coefficients. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *29*(2), 297–301.

Sohn, Hoon, Czarnecki, Jerry A., & Farrar, Charles R. (2000). Structural health monitoring using statistical process control. *Journal of the Structural Engineering*, *126*(11), 1356–1363.

Uddin, Zahoor, Ahmad, Ayaz, Qamar, Aamir, & Altaf, Muhammad (2018). Recent advances of the signal processing techniques in future smart grids. *Human-Centric Computing and Information Sciences*, *8*(1), 1–15.

Vellido, Alfredo, Martín-Guerrero, José David, & Lisboa, Paulo J. G. (2012). Making machine learning models interpretable. In *Proc. of the 2012 European symposium on artificial neural networks* (pp. 163–172).

Verleysen, Michel, & François, Damien (2005). The curse of dimensionality in data mining and time series prediction. In *Proc. of the 8th international work-conference on artificial neural networks* (pp. 758–770).

Vishwanathan, Vichy S. V., Smola, Alexander J., & Vidal, René (2007). Binet–Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. *International Journal of Computer Vision*, *73*(1), 95–119.

Welch, Peter D. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, *15*(2), 70–73.

Werner, Rolf (2012). Sunspot number prediction by an autoregressive model. *Sun and Geosphere*, *7*(2), 75–80.

Wolf, Lior, & Shashua, Amnon (2003). Learning over sets using kernel principal angles. *Journal of Machine Learning Research*, *4*, 913–931.

Yule, George Undy (1927). On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London Series A*, *226*, 267–298.

Oliver Lauwers (°1991, Ph.D. (2020) KU Leuven, Belgium) is a free researcher at the Department of Electrical Engineering at the KU Leuven (Leuven, Belgium). Supported by an IWT/FWO Strategic Basic Research Ph.D. Fellowship, he received a doctoral degree in Electrical Engineering there in 2020, on the topic of Time Series Clustering, with a focus on interpretable distance measures for time series. During this time, he also completed the InnoEnergy Ph.D. School on innovation in the energy sector. He is the founder of Cepsis, a research consultancy firm specialized in time series analysis in the energy industry.

Christof Vermeersch (°1994) is a doctoral student at the Department of Electrical Engineering at the KU Leuven (Leuven, Belgium), supported by an FWO Strategic Basic Research Ph.D. Fellowship. As part of the research group led by Bart De Moor, he studies system identification techniques and numerical linear algebra algorithms to solve systems of multivariate polynomial equations and multiparameter eigenvalue problems. He received a master degree in Mathematical Engineering in 2017 from the same university.

Bart De Moor (°1960, Ph.D. (1983) KU Leuven, Belgium) was a research associate at Stanford University, is now full Professor at the Department of Electrical Engineering of KU Leuven and a guest professor at the Universita di Siena, Italy. He served several times as head of cabinet of ministers of Science and Socio-Economic Policy in Belgium/Flanders and acted as a vice-rector of International Policy of KU Leuven. His fields of research are numerical linear algebra, system theory and control, machine learning and data science, in which he has been guiding more than 80 Ph.D. students and co-authored more than 400 scientific papers and 11 books.

Dr. De Moor has been serving in numerous international scientific and science policy assessment committees, is member/head of the board of several (inter-)national scientific institutes (including the Flanders Biotech Institute) and funding agencies. He co-founded 8 spin-off companies (4 in Health 2.0, 4 in Industry 4.0). Since 2019, he is one of the architects and coordinators of the large Flanders AI program. He is the chairman of the Capricorn Digital Growth Fund (venture capital), of Health-House (a high-tech science outreach centre) and the Alamire Foundation (digital humanities, polyphonic music) and co-founded Technopolis (children's science center).

A member of the Royal Academy, he was awarded with numerous scientific prizes and recognitions (a.o. an ERC Advanced Grant) and honors (Fellowships of IEEE, SIAM). In 2010 he received the Science Excellence Award from King Albert II of Belgium and in 2020 was nominated a Commander in the Order of Leopold, by King Filip I.