A Time Series Distance Measure for Efficient Clustering of Input/Output Signals by Their Underlying Dynamics

Oliver Lauwers and Bart De Moor

***Abstract*—Starting from a dataset with input/output time series generated by multiple deterministic linear dynamical systems, this letter tackles the problem of automatically clustering these time series. We propose an extension to the so-called Martin cepstral distance, that allows to efficiently cluster these time series, and apply it to simulated electrical circuits data. Traditionally, two ways of handling the problem are used. The first class of methods employs a distance measure on time series (e.g., Euclidean, dynamic time warping) and a clustering technique (e.g., *k*-means, *k*-medoids, and hierarchical clustering) to find natural groups in the dataset. It is, however, often not clear whether these distance measures effectively take into account the specific temporal correlations in these time series. The second class of methods uses the input/output data to identify a dynamic system using an identification scheme, and then applies a model norm-based distance (e.g., H_2 and H_∞) to find out which systems are similar. This, however, can be very time consuming for large amounts of long time series data. We show that the new distance measure presented in this letter performs as good as when every input/output pair is modeled explicitly, but remains computationally much less complex. The complexity of calculating this distance between two time series of length N is $\mathcal{O}(N \log N)$.**

***Index Terms*—Pattern recognition and classification, machine learning, linear systems.**

Manuscript received March 5, 2017; revised May 15, 2017; accepted June 9, 2017. Date of publication June 14, 2017; date of current version June 26, 2017. This work was supported in part by the Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical Systems, Control and Optimization, 2012–2017) Flemish Government: IWT: TBM IETA under Grant 130256, in part by the Ph.D. grants Industrial Research fund (IOF): IOF Fellowship 13-0260 VLK Stichting E. van der Schueren: Rectal Cancer EU H2020-SC1-2016-2017 under Grant 727721, in part by MIDAS Meaningful Integration of Data, Analytics and Services KU Leuven Internal Funds under Grant C16/15/059 and Grant C32/16/013, and in part by KIC EIT Health: New MOOC—Data Analytics in Health; EIT Health Summer School Innovation on Big Data for Healthy Living imec strategic funding 2017. The work of O. Lauwers was supported by the SB-Grant of the FWO (formerly IWT) under Grant 10.13039/501100003130. Recommended by Senior Editor G. Cherubini. *(Corresponding author: Oliver Lauwers.)*

The authors are with the Department of Electrical Engineering, 3000 Leuven, Belgium, and also with IMEC, Leuven, Belgium (e-mail: oliver.lauwers@esat.kuleuven.be; bart.demoor@esat.kuleuven.be).

Digital Object Identifier 10.1109/LCSYS.2017.2715399

I. Introduction

TIME series clustering is an important topic in modern research. State-of-the-art clustering methods of other data types are often not suited for this high-dimensional, temporally correlated data structure. Clustering is the task of finding groups with similar elements in a dataset and consists of three components: a similarity measure based on relevant data features, a clustering algorithm and an evaluation criterion. While the latter two components might carry over, defining a good distance measure is a difficult problem, especially if one is interested in the dynamics of the generating dynamical system of the time series.

Representing the time series as convolutional single-input single-output (SISO) linear time invariant (LTI) deterministic dynamical systems further generates problems of its own, as the contributions of the input signal and the impulse response of the system are convolved in the time domain. It is thus not intuitively clear how these two contributions can be separated, for example when one is interested only in the dynamics of the system and not in the specific input signal.

This problem grows ever more relevant as large scale big data time series problems grow more prevalent in areas like finance, medicine, or the industrial Internet of Things, where clustering is important in tasks like anomaly detection [7], [13]. A typical industrial problem contains several hundred sensors per machine, tens of machines per plant, and several plants per industrial player, collecting data every few seconds, for months or even years of operation time. This results in datasets of several million time points for thousands of series. Clustering techniques should thus scale well.

In Section II we look at state-of-the-art clustering methods for time series from two perspectives, starting from a dataset containing input/output time series pairs, generated by different SISO LTI dynamical systems. From a machine learning point of view, we use an automated clustering method with an off-the-shelf time series distance such as the Euclidean distance or Dynamic Time Warping (DTW). From a system identification point of view, we apply norms such as the H_2 or H_∞ norm to compare systems estimated from the data. We find that these techniques either are very fast, but give poor results, or perform well, but are computationally expensive.

Next, in Section III, we look at the Martin cepstral distance [3], [8], which combines insights from systems theory into a distance measure that can be computed on the raw data. This metric was defined for SISO ARMA models (i.e., LTI models that use white noise as an input signal).

The main contribution of this letter is an extension of the cepstral distance measure, that incorporates deterministic input signals, and allows to calculate distances between a broader class of SISO LTI dynamical systems. It thus allows to cluster time series by dynamics, but remains computationally much simpler than explicitly estimating models.

Subsequently, we apply this new distance measure in Section IV to a simulation of electrical circuits, where we generate a dataset consisting of input/output signal pairs, and the problem is to identify which data belong to which generating system. Finally, we conclude this letter and provide some paths for future research in Section V.

II. Existing Methods

Existing methods to cluster time series employ a clustering technique, together with some distance measure. Liao [6] discerns three types of distance measures: measures based on raw data, measures based on features of the time series and measures based on models. For the scope of this letter, we will focus on the first and the latter (as the distance measure we propose combines elements of these two broad classes). We present two raw data distance measures, the Euclidean metric and the Dynamic Time Warping metric [5], and two model-based distance measures, connected to the H_2-norm and the H_∞-norm. In the next section, we will introduce and extend the cepstral distance [3], [8], which combines the efficiency of the raw data distance measures with the insight in generative dynamics of the model norms, and thus has representations both as a raw data distance and as a model-based one.

A. Raw Data Distance Measures

In what follows we will define u_m to be the input signal of the m-th element of a dataset, y_m is the corresponding output signal and $u_m(k)$ or $y_m(k)$ is the value at timepoint k of respectively the input and output of the m-th element of the input/output dataset. Time series from element m start at $k = 0$ and end at $k = N_m$. The system that generated an output from a given input will be called the generating (dynamical) system.

1) Euclidean Distance:

Definition 1: The Euclidean distance, $d_E(\cdot, \cdot)$ treats the time series as a vector, and applies the element-wise Euclidean vector distance between two time series of same length N_m, defined as

$$d_E(y_m, y_n) = \sqrt{\sum_{k=0}^{N_m} (y_m(k) - y_n(k))^2}. \tag{1}$$

Advantages

- The Euclidean distance is easy to calculate, allowing for very efficient computation and clustering.
- No system identification step is needed.

Disadvantages

- There is no clear link between this distance measure and the generating system.
- This measure treats the time series as a vector, and ignores the temporal correlations in the data.
- This measure does not allow to compute distances between time series of different lengths.
- This measure does not take the input into account.

2) Dynamic Time Warping:

Dynamic Time Warping (DTW) [5], [12] is an algorithm that tries to locally align time series, by *warping* them such that the Euclidean distance between the warped time series is minimal. Mathematically, this *warping*, and the measure that is found in this way, can be described as follows.

Given two output signals, y_1 and y_2, of length N_1 and N_2 respectively, a matrix M is constructed, where the (l, m)-th element of M is defined as $M_{(l,m)} = (y_1(l) - y_2(m))^2$. A warping path, $W = w_1, w_2, \ldots, w_k, \ldots, w_K$ is then defined, with each $w_k = (M_{(l,m)})_k$ an element of matrix M and $max(N_1, N_2) \leq K < N_1 + N_2 - 1$.

The path is subject to the boundary conditions $w_1 = M_{1,1}$ and $w_K = M_{N_1,N_2}$ (i.e., the path starts in one corner of the matrix and ends in the opposite one), has to be continuous, in such a way that two consecutive elements w_k and w_{k+1} are maximally one column and one row apart, and has to be monotonously increasing in its indices, i.e., that in going from w_k to w_{k+1}, column nor row number can decrease.

Definition 2: We are now interested in the warping path W_{DTW} that minimizes the cost function

$$d_{DTW}(y_1, y_2) = min\left\{\sqrt{\sum_{k=1}^{K} w_k}\right\}. \tag{2}$$

The sum over this path is then the DTW distance between the time series.

Though this algorithm is computationally expensive due to the combinatorial nature of the problem, several lower bounds have been devised that can be implemented efficiently. In what follows, we use the lower bound from [5].

Advantages

- The DTW distance takes into account (part of) the local temporal correlations.
- No system identification step is needed.
- Lower bounds on the distance are reasonably efficient.
- This measure allows to calculate distances between time series of different lengths.

Disadvantages

- There is no clear link between this distance measure and the generating system.
- The DTW distance as such is expensive to calculate.
- This measure does not take the input into account.

B. Model-Based Distance Measures

We use the same notation as in Section II-A. The generating system of the input/output pair (u_m, y_m) will be denoted by M_m, and its corresponding transfer function will be written $\mathcal{H}_m$. Based on a model norm $\|\cdot\|$, the distance between two models M_i and M_j is defined as $\|\mathcal{H}_i - \mathcal{H}_j\|$.

1) H_2-Norm:

Definition 3: The H_2-norm, $\|\mathcal{H}\|_2$, of a discrete-time system M with transfer function $\mathcal{H}$ is defined as

$$\|\mathcal{H}\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi} Tr\left\{\mathcal{H}^H(e^{i\omega})\mathcal{H}(e^{i\omega})\right\} d\omega}, \tag{3}$$

where $Tr\{\}$ denotes the trace, the superscript $^{.H}$ denotes the Hermitian conjugate and i denotes the imaginary unit.

The H_2-norm can be seen as the root-mean-square of the system response to a normalized white noise input. It is thus a measure of the power, or steady-state variance of this response. The H_2-norm will be infinite for unstable systems.

Advantages

- The H_2-norm provides a physically interpretable way to characterize underlying dynamics of time series.
- This norm allows to calculate distances between time series of different lengths.
- This norm takes the input data into account.

Disadvantages

- A system identification procedure is needed, which is both difficult to automate and often computationally expensive (at least more expensive than the raw data measures).

2) H_∞-Norm:

Definition 4: The H_∞-norm, $\|\mathcal{H}\|_\infty$, of a discrete-time system M with transfer function $\mathcal{H}$ is calculated as

$$\|\mathcal{H}\|_\infty = \max_{\omega\in[0,\pi[} |\mathcal{H}(e^{i\omega})|. \tag{4}$$

This norm thus measures the maximal gain of the frequency response and is called the *gain* of the system. It becomes infinite for systems with poles on the unit circle.

Advantages

- The H_∞-norm provides a physically interpretable way to characterize underlying dynamics of time series.
- This norm allows to calculate distances between time series of different lengths.
- This norm takes the input data into account.

Disadvantages

- A system identification procedure is needed, which is both difficult to automate and often computationally expensive (at least more expensive than the raw data measures).

III. Cepstral Distance

In this section we take a closer look at an insightful distance measure on ARMA models, which can be interpreted both as a raw data distance measure and as a model norm: the Martin cepstral norm [3], [8]. We first give a very concise review of the cepstral norm in the stochastic case, then proceed with an extension that allows us to incorporate information about the deterministic input signal.

A. Original Cepstral Norm

Based on the power spectral density, Φ_y, of a signal y, we can define its power cepstrum, c_y as

$$c_y = \mathcal{F}^{-1}(\log(\Phi_y)), \tag{5}$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. This produces a series of coefficients, $c_y(k)$, with integer $k \in [0, N]$, where N denotes the length of time series y.

Definition 5: The cepstral norm, $\|\mathcal{H}\|_C$, of model M with transfer function $\mathcal{H}$, and output y is defined as

$$\|\mathcal{H}\|_C = \sum_{k=0}^{N} k\big(c_y(k)\big)^2. \tag{6}$$

For ARMA models it was proven that there are multiple methods to calculate this norm: it can be derived from the subspace angles of the output Hankel matrices of the generating system [3] or from the mutual information of the output space of a system, and from a combination of poles and zeros of the transfer function of the model [8] (equivalent to the one we will derive in Section III-B). Moreover, equation (6) allows us to calculate the norm straight from raw data (see also [9, Ch. 10]), without the need to identify the underlying systems. We can thus connect the cepstral norm to a raw data distance measure in the following sense.

Definition 6: The cepstral distance, $d_C(y_i, y_j)$, between two time series, y_i and y_j, is defined as

$$d_C(y_i, y_j) = \sum_{k=0}^{\max\{N_i,N_j\}} k\big(c_{y_i}(k) - c_{y_j}(k)\big)^2, \tag{7}$$

where $\max\{N_i, N_j\} - \min\{N_i, N_j\}$ zeros are added at the end of the cepstrum of length $\min\{N_i, N_j\}$.

Advantages

- The cepstral distance has an interpretation in terms of the generating models of the time series.
- The cepstral distance is easy to calculate, allowing for very efficient computation and clustering.
- No system identification step is needed.
- This measure allows to calculate distances between time series of different lengths.

Disadvantages

- This distance measure can only take information coming from a stochastic input into account.

B. Extended Cepstral Distance

The cepstrum, defined in the previous section, finds its roots in homomorphic signal processing [9, Ch. 10]. In this type of processing, the original time series data, which often involves complex multiplicative operations like convolutions, is mapped, through a non-linear mapping, to a different domain, that allows for linear filtering. The cepstrum, as in equation (5), is a good example. The convolution in the time domain changes into a multiplication by calculating the power spectral density:

$$\Phi_y = |\mathcal{H}|^2\Phi_u. \tag{8}$$

Applying a logarithmic transformation then turns the multiplication in frequency domain into an addition, and we get

$$\log(\Phi_y) = \log(|\mathcal{H}|^2) + \log(\Phi_u). \tag{9}$$

Finally, the inverse Fourier transform takes the problem back to (a transformed version of) the time domain. Equation (5) is thus effectively a method to transform the convolution into an addition. Defining the cepstrum coefficients of the input signal u as $c_u(k)$, and the contribution to the cepstrum coefficients of the transfer function $\mathcal{H}$ as $c_h(k)$, we can write

$$c_y(k) = c_u(k) + c_h(k). \tag{10}$$

This allows us to take the output, and separate the contributions from the input signal (which was the main disadvantage left in the cepstral distance, see Section III-A) and the impulse responses of the system. Based on input/output signal pairs,

we now have a measure of the underlying generating system dynamics by looking at $c_h(k) = c_y(k) - c_u(k)$.

Definition 7: The extended cepstral distance, $d_{C_e}((y_i, u_i), (y_j, u_j))$, between two input/output pairs of time series, (y_i, u_i) and (y_j, u_j), with respective transfer functions $\mathcal{H}_i$ and $\mathcal{H}_j$, is defined as

$$d_{C_e}((y_i, u_i), (y_j, u_j)) = \sum_{k=0}^{\min\{N_i, N_j\}} k\big(c_{h_i}(k) - c_{h_j}(k)\big)^2. \tag{11}$$

By interpreting the transfer function as a quotient of two polynomials, with zeros and poles as their respective roots, we can find an interpretation of these cepstrum coefficients. Following the reasoning in [9, Ch. 10], we find (for stable, minimum-phase systems):

$$c_h(k) = \sum_{j=1}^{p} \frac{\alpha_j^{|k|}}{|k|} - \sum_{j=1}^{q} \frac{\beta_j^{|k|}}{|k|} \quad \forall k \neq 0, \tag{12}$$

where the α's denote the p poles, and the β's denote the q zeros. This expression (which is analogous to the one in the stochastic case) sets us on the path to rederive the framework presented in [3] for the deterministic case. This, however, requires an extension of the notion of subspace angles, which is out of the scope of this letter.

We propose this extended cepstral distance as a way to efficiently cluster input/output data by generating dynamics.

Advantages

- The extended cepstral distance is linked to the generating model of the time series.
- The extended cepstral distance is easy to calculate, allowing for very efficient computation and clustering.
- No system identification step is needed.
- This measure allows to calculate distances between time series of different length.
- This measure takes the input into account.

Disadvantages

- The interpretation of the measure in terms of system parameters and properties is not immediately clear, thus the theoretical framework of the original cepstral distance does not carry over trivially.

C. Computational Overview

A pseudo-code overview of the algorithm is shown in Algorithm 1. Calculating the extended cepstral distance amounts to estimating the power spectral density of both input and output by Welch's method [14] (employing the FFT,[1] which is of $\mathcal{O}(n \log n)$, with n the length of the windows considered in Welch's method), taking the logarithm of the resulting vector, and then applying an inverse Fourier transform (employing the IFFT, running in $\mathcal{O}(N \log N)$ time, with N the length of the time series) on them. In the end, we then apply a weighted Euclidean distance on the results.

Note that, for very short time series (less than 2^7 time points), we use the multitaper method [10] to achieve better results. This method is a bit slower than Welch's method, but because we only apply it in the case of short time series, this does not matter too much for the analysis here. The complexity of calculating the extended cepstral distance between two time series is then $\mathcal{O}(N \log N)$, with N the length of the time series. That of the Euclidean distance is $\mathcal{O}(N)$, standard DTW is $\mathcal{O}(N^2)$, while the lower bound in [5] is also $\mathcal{O}(N \log N)$. The complexity of explicitly identifying a system and calculating model norms depends very much on the identification method used and the model size. Many of these techniques employ, at their core, a SVD decomposition, of order $\mathcal{O}(M^2 N)$, with M here being the model order. Often this core is then repeated in non-convex optimization problems, solved iteratively. Furthermore, hyperparameters (e.g., model order) need to be estimated, further increasing the computational cost. For example, a simple grid search repeats this whole process for every point in the grid.

Algorithm 1: Algorithm for the Extended Cepstral Distance

input : Two input/output signal pairs, (y_1, u_1) of length N_1, and (y_2, u_2) of length N_2

output: The extended cepstral distance $d_{C_e}((y_1, u_1), (y_2, u_2))$ between these two pairs, as defined in Subsection III-B

for $i \leftarrow 1$ **to** 2 **do**
  $\Phi_{u_i} \xleftarrow{\text{Welch's Method}^3} u_i$
  $c_{u_i} \leftarrow \text{ifft}\big(\log\big(\Phi_{u_i}\big)\big)$
  $\Phi_{y_i} \xleftarrow{\text{Welch's Method}} y_i$
  $c_{y_i} \leftarrow \text{ifft}\big(\log\big(\Phi_{y_i}\big)\big)$
  // c_{u_i} and c_{y_i} are vectors of length N_i
end
$w = \big[0, 1, \ldots, \max\{N_1, N_2\} - 1\big]$
add $(\max\{N_1, N_2\} - \min\{N_1, N_2\})$ 0's to the cepstra of the signal pair of length $\min\{N_1, N_2\}$
$d_{C_e}((y_1, u_1), (y_2, u_2)) \leftarrow w * \big(\big(c_{y_1} - c_{u_1}\big)^\intercal - \big(c_{y_2} - c_{u_2}\big)^\intercal\big)^2$

IV. Application on Electrical Circuits

A minimal working example of the simulations performed in this Section is available on GitHub.[4]

A. Simulation Set-Up

To test the proposed techniques, we simulate data coming from electrical circuits. We start out by modelling two circuits with the same topology, but different values for the R, L, and C components. The topology was taken from a course on linear physical systems analysis [2]. The network topology and the values of the components are shown in Figure 1. The input of the system is the current i_u, the output is the voltage over L_2, e_y. State-space models of order 3 are then written down for these networks.

We provide both systems with 200 different input signals (100 outputs of LTI models of order 15, 50 multisine waves corrupted by Gaussian white noise with standard deviation of 0.1 and 50 white noise signals), and measure the outputs. This generates a dataset of 400 input/output signal pairs (200 inputs times 2 models). The question at hand is whether we can use only this input/output data, to determine which pairs were generated by the same system, i.e.,

[1]Note that, for longer time series (i.e., 2^{10} and beyond), the Fast Fourier Transform [1] provides a clean enough output to work on. We could thus speed up the algorithm even further for longer series.

[3]For very short time series (i.e., fewer than 2^7 points), we use the multitaper method [10] to achieve better results.

[4]https://github.com/Olauwers/Extended-Cepstral-Distance

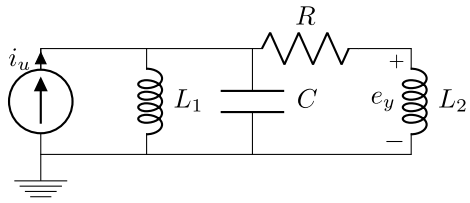

Fig. 1. Electric circuit that was used for the experiments. Two sets, S_1 and S_2, of values were chosen for the components, namely $S_1 = \{R = 100\Omega, L_1 = 60\text{H}, L_2 = 20\text{H}, C = 50\text{F}\}$ and $S_2 = \{R = 100\Omega, L_1 = 160\text{H}, L_2 = 200\text{H}, C = 75\text{F}\}$. These two electrical circuits were used to perform the simulations in Section IV.

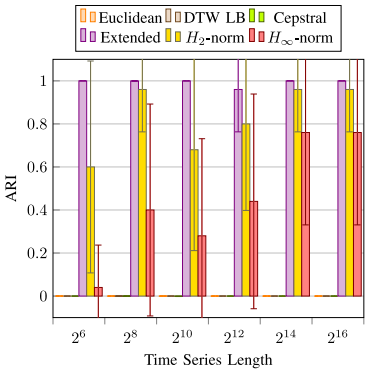

Fig. 2. Performance of the different clustering algorithms, as measured by the ARI. For each time series length, shown on the x-axis, the average ARI over 100 experiments of finding 2 clusters in 400 time series is depicted as the height of the bar. The error bars show the standard deviation for the performance on these 100 experiments. Note that the Euclidean, DTW LB and cepstral distance have an ARI of 0, i.e., they amount to random guessing. The extended cepstral distance performs best for all series lengths. The model based distances were given a wrong model order, but still give good performance for longer time series.

cluster the dataset in two groups, defined by the generative dynamics.

We will do this using the distance measures from Sections II and III-A, keeping in mind that we use the lower bound (DTW LB) from [5] as an efficient lower bound to DTW. We compare to the technique developed in Sections III-B and III-C, where we respectively gave a theoretical and computational overview of the distance measure.

The performance of these simulations will be measured by the Adjusted Rand Index (ARI) [4], [11], which is a similarity measure between partitions. The ARI compares two partitions, S_1 and S_2, by calculating the ratio of pairs that have the same partitioning status (i.e., belonging to the same partition or not) in both S_1 and S_2 to the total amount of data pairs, then adjusting the resulting ratio by subtracting the expected value, to account for guessing (i.e., a partitioning that is the result of random guessing is assigned an ARI of 0). An ARI of 1 corresponds to perfectly similar partitions.

We compare the partitions generated by a hierarchical clustering method, cut-off at two clusters, using distance matrices generated by the different distance measures of Sections II and III versus the ground truth (i.e., the time series was generated by the system with parameters S_1 or with parameters S_2, as in Figure 1).

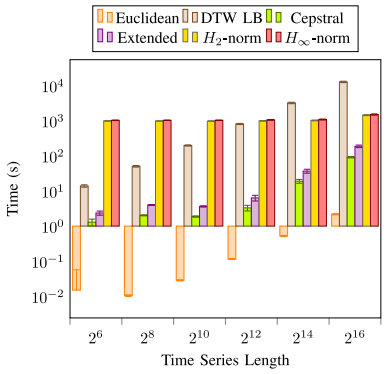

Fig. 3. Execution time of the different clustering algorithms, measured in seconds. For each time series length, shown on the x-axis, the average time over 100 experiments of finding 2 clusters in 400 time series is depicted as the height of the bar. The error bars show the standard deviation for the execution time on these 100 experiments. Note that the y-axis has logarithmic scale. The extended cepstral distance remains several orders of magnitudes faster than the model-based distances. Note that DTW LB quickly becomes the computationally most expensive technique. The Euclidean distance is always fastest.

B. Results

The results for the set-up in the previous subsection are shown in Figure 2, which shows the average and standard deviation for the ARI of the simulation results, and Figure 3, which shows the average and standard deviation for the execution time of the simulations.

The extended cepstral distance gives the best results, managing to cluster the simulated signal pairs almost perfectly every time. This is to be expected, as it was tailored specifically to take into account the dynamics of the underlying model,[5] and nothing but those dynamics. The reasons why it performs better than the other measures will be explained in what follows, and we again use the distinction between raw data and model-based distances measures from Section II.

1) Raw Data Distance Measures: The reason the other raw data distance measures do not perform well on the problem at hand, is because they are not able to separate the input signal from the output signal. The dynamics of the output are dominated by the input (the models generating the inputs are of higher order than the models describing the electrical circuits). If we only use white noise inputs the original cepstral distance performs better (see the left hand side in Figure 4).[6] Euclidean and DTW distances still do not deliver good results when detecting the difference in dynamics. Explicitly adding in the distance between the inputs did not improve results, which were omitted.[7]

There is no hope to achieve better results by taking the input signal into account in the Euclidean distance or the DTW distance, as these distances look at the shape of the signal – which does not solve the problem at hand –, but at its generative dynamics. DTW is better at this job [5], but has a big disadvantage: it takes a lot of time to compute, especially

[5]We redid the experiments for generating systems of higher order, and the extended cepstral distance still performed best. Results were omitted.

[6]The original and extended cepstral distance are equivalent in this case, as the cepstrum of white noise is only non-zero in its zeroth component, which is omitted in the sum in equations (7) and (11), which then coincide.

[7]The code is still available on GitHub. The reader is encouraged to try out any adaptation to the distance measures presented.

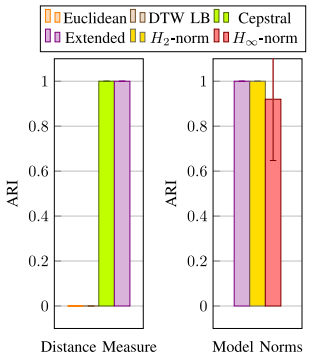

Fig. 4. On the left, the performance is shown of the different raw data distance measures, as measured by the Adjusted Rand Index (ARI), in the case of white noise as an input, and time series of length 2^{10}. Here, the average over 100 experiments with 400 output signals is shown. Note that the original cepstral distance now shows the same performance as the extended one. On the right, results of an experiment where we provided the system identification step with the correct orders of the models are shown. Here, we calculated an average over 100 experiments with 40 output signals, to reduce computation time. Again, we simulated time series of length 2^{10}. The model-based distances now show better performance.

for long time series, where it even surpasses the model-based distance measures in computation time (see Figure 3).

The extended cepstral distance is thus preferred to cluster input/output signals by dynamics of their generating models.

2) Model-Based Distance Measures: The model-based distance measures show better results than the raw data distance measures, and this again is to be expected. The model-based measures manage to peel out the information on the system that generated the input/output pair. However, a priori we have no information on the order of the underlying system, so we arbitrarily have to set a model order. In this case, we estimated transfer functions of order 5. If we share the correct model order (3) with the identification algorithm, performance of the model norms increases (Figure 4).

Schemes exist to estimate model orders, and more effort can be put in correctly identifying the underlying model. However, the model norm techniques are already several orders of magnitude slower than the extended cepstrum distance measure (Figure 3). For problems concerning large numbers of long input/output-pairs, as can be found in realistic problems in process industry (see, for example, [7], where more than 250 sensors make a measurement every 5 minutes for 6 months), this becomes highly impractical.

The extended cepstral distance is thus preferred over explicitly identifying systems, because of both being easier to automate, and taking less time to compute.

C. Non-Linear Loads

We performed additional tests on robustness against nonliner loads in the circuits, replacing constant inductors with saturating ones. Results depend heavily on saturation levels and signal length, but the extended cepstral distance still outperforms the others. We therefore do not show results here, but provide an interactive tutorial on the accompanying GitHub, where the reader is encouraged to experiment with different values for the parameters of the non-linearity.

V. Conclusion

We present a distance measure that is as insightful as a model norm-based distance, yet remains computationally much simpler than explicitly estimating models. It allows to meaningfully cluster large input/output signal pair datasets based exclusively on the dynamics of the generating systems. We have tested it on a simulation of electrical circuits, where we started from two circuits with a current as input and a voltage difference over an inductor as output. We provided both circuits with 200 different inputs, resulting in 400 input/output pairs. The proposed measure performs as well as model-based distances on estimates of the generative systems, but is much easier to calculate. Other distance measures (Euclidean, DTW) perform much worse.

We furthermore show that, in the stochastic input case, the extended distance proposed in this letter reduces to the original cepstrum distance, which was proven [3], [8] to be equivalent to a model norm. This gives hope that the extended distance could also be linked to a model norm.

The results indicate the extended cepstral distance measure does a good job of capturing the dynamics of input/output pairs. An application to a real-life dataset is needed to validate the effectiveness in practice, but for the simulated problem at hand, the distance measure succeeded in perfectly distinguishing different dynamics based on raw data alone.

Further research should look at further extensions of the distance measure to more general classes of systems, such as nD systems and MIMO systems. It is not intuitively clear how the cepstrum should be defined in these cases, which necessitates further effort.

References

[1] E. O. Brigham, *The Fast Fourier Transform and Its Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.
[2] E. Cheever, "Linear physical systems analysis," Dept. Eng., Swarthmore College, Swarthmore, PA, USA, Apr. 2016. [Online]. Available: http://lpsa.swarthmore.edu/
[3] K. De Cock and B. De Moor, "Subspace angles between ARMA models," *Syst. Control Lett.*, vol. 46, no. 4, pp. 265–270, 2002.
[4] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, 1985.
[5] E. Keogh, "Exact indexing of dynamic time warping," in *Proc. 28th Int. Conf. Very Large Data Bases*, Hong Kong, 2002, pp. 406–417.
[6] T. W. Liao, "Clustering of time series data—A survey," *Pattern Recognit.*, vol. 38, no. 11, pp. 1857–1874, 2005.
[7] L. Martí, N. Sanchez-Pi, J. M. Molina, and A. C. B. Garcia, "Anomaly detection based on sensor data in petroleum industry applications," *Sensors*, vol. 15, no. 2, pp. 2774–2797, 2015.
[8] R. J. Martin, "A metric for ARMA processes," *IEEE Trans. Signal Process.*, vol. 48, no. 4, pp. 1164–1170, Apr. 2000.
[9] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ, USA: Pearson, 1975.
[10] D. B. Percival and A. T. Walden, *Spectral Analysis for Physical Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1993.
[11] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.
[12] C. A. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," in *Proc. 3rd Workshop Min. Temporal Sequential Data*, Seattle, WA, USA, 2004.
[13] T. Vafeiadis *et al.*, "Robust malfunction diagnosis in process industry time series," in *Proc. IEEE 14th Int. Conf. Ind. Informat. (INDIN)*, Poitiers, France, Jul. 2016, pp. 111–116.
[14] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, Jun. 1967.