

# Constrained Dynamic Mode Decomposition

Tim Krake, Daniel Klötzl, Bernhard Eberhardt, and Daniel Weiskopf, *Member, IEEE*

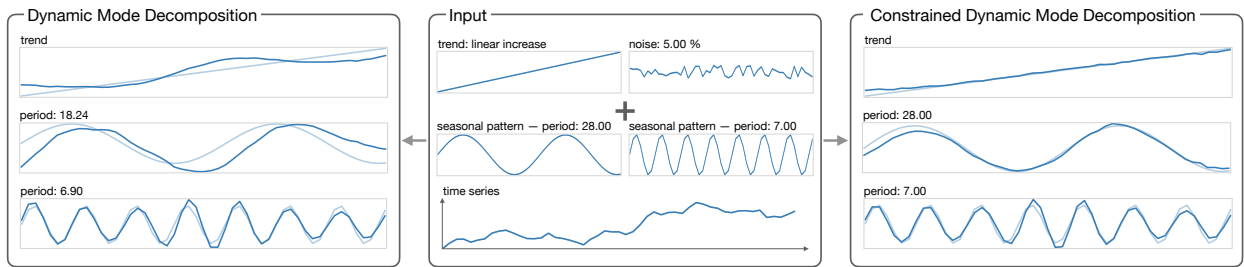


Fig. 1. Application of DMD and constrained DMD to an artificial time series that consists of four different patterns: linear trend, two seasonal patterns with the periods 7 and 28, and noise. Thus, the superposed time series is a typical example of daily data exhibiting weekly and monthly patterns. While DMD detects the weekly pattern with an identified period of 6.90, it fails to compute the correct trend and monthly pattern. Incorporating both frequencies, our constrained DMD identifies all patterns correctly.

**Abstract**— Frequency-based decomposition of time series data is used in many visualization applications. Most of these decomposition methods (such as Fourier transform or singular spectrum analysis) only provide interaction via pre- and post-processing, but no means to influence the core algorithm. A method that also belongs to this class is Dynamic Mode Decomposition (DMD), a spectral decomposition method that extracts spatio-temporal patterns from data. In this paper, we incorporate frequency-based constraints into DMD for an adaptive decomposition that leads to user-controllable visualizations, allowing analysts to include their knowledge into the process. To accomplish this, we derive an equivalent reformulation of DMD that implicitly provides access to the eigenvalues (and therefore to the frequencies) identified by DMD. By utilizing a constrained minimization problem customized to DMD, we can guarantee the existence of desired frequencies by minimal changes to DMD. We complement this core approach by additional techniques for constrained DMD to facilitate explorative visualization and investigation of time series data. With several examples, we demonstrate the usefulness of constrained DMD and compare it to conventional frequency-based decomposition methods.

**Index Terms**—Dynamic Mode Decomposition, time series analysis, spectral decomposition, frequency-based constraints, human-in-the-loop

## 1 INTRODUCTION

The explorative visualization and analysis of time series is a wide-ranging field of research. It includes, for instance, the detection of anomalies, the identification of seasonal and cyclic patterns, and forecasting. Nowadays, explorative visualizations support these tasks via a combination of various tools based on statistics, signal processing, or machine learning. In general, these tools allow only little (or even no) interaction with their core analysis mechanisms. This is also the case for common spatio-temporal decomposition techniques that result in a characteristic representation with modes. One of these techniques is Dynamic Mode Decomposition (DMD), based on spectral theory and mainly used in the field of dynamical systems and fluid dynamics.

In this paper, our approach is to incorporate frequency-based constraints into DMD to obtain an adaptive decomposition. This leads to *constrained Dynamic Mode Decomposition (constrained DMD)*, which directly addresses the problem of facilitating user interaction with the core mechanisms. The effect of frequency-based constraints is shown

in Fig. 1, where the time series describes stereotypical daily data with weekly and monthly patterns. The length of the time series makes an identification of the monthly pattern difficult. This is a common problem for time series with multiple time scales. Constrained DMD solves this issue due to the incorporation of user-defined frequencies. Hence, our main contributions can be summarized as follows:

- An equivalent reformulation of DMD is presented that enables incorporation of multiple frequency-based constraints.
- We provide an efficient algorithm that uses the frequency-based constraints in a human-in-the-loop manner.
- We present additional visualization techniques that complement the visual exploration and analysis.

In addition to our technical contributions, we demonstrate the usefulness of our approach with regard to two aspects: First, a qualitative and quantitative comparison to other common spatio-temporal decomposition techniques is provided. Second, our approach is evaluated with focus on the interaction with the frequency-based constraints.

## 2 RELATED WORK

There are numerous approaches in the literature that address the analysis of data via explorative visualization. An overview of general interactive visualization techniques is presented by Tominski and Schumann [53]. For time-oriented data, an overview is given by Aigner et al. [1]. Here, an important challenge is the identification of trend, seasonal, and cyclic patterns. This is also one of the main goals of this paper. For this task, we introduce a formulation of DMD that enables interaction with the core mechanism in a human-in-the-loop manner. Thus, the related work focuses on two aspects: spatio-temporal decompositions (such as DMD) and time-oriented visualization techniques with similar goals

- *Tim Krake is with University of Stuttgart and Hochschule der Medien. E-mail: tim.krake@visus.uni-stuttgart.de.*
- *Daniel Klötzl is with University of Stuttgart. E-mail: daniel.kloetzl@visus.uni-stuttgart.de.*
- *Bernhard Eberhardt is with Hochschule der Medien. E-mail: eberhardt@hdm-stuttgart.de.*
- *Daniel Weiskopf is with University of Stuttgart. E-mail: daniel.weiskopf@visus.uni-stuttgart.de.*

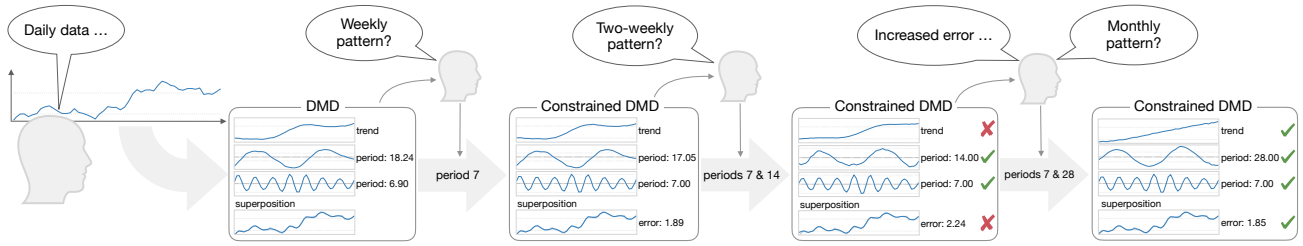


Fig. 2. A user perspective step-by-step application of our approach. The schematic visualization shows multiple application of our constrained DMD in a human-in-the-loop manner. The user performs feedback-loops on the basis of external knowledge or present observations.

that integrate the human-in-the-loop. In the following, we start with a comparison to DMD-based approaches followed by more general ones.

Many theoretical and algorithmic concepts of DMD [45, 46] were developed in the literature and applied to time series data, particularly in the field of dynamical systems and fluid dynamics. While these approaches address either the efficiency [9, 20] or the accuracy of identified frequencies [14, 41, 54], they do not provide any mechanism to directly influence the core algorithm with respect to the computation of frequencies; we want to fill this gap with this paper.

More closely related is DMD with control (DMDc) [43], a DMD variant that integrates external control to the dynamics. Therefore, it is possible to analyze complex systems that require additional control. However, DMDc addresses completely different problems, whereas our technique incorporates constraints that directly influence DMD.

Another strategy that is closely related to our approach is constraining the DMD-specific minimization problem. Inspired by the Perron-Frobenius operator (the dual to the Koopman operator), Goswami et al. [23] introduce a minimization problem with constraints that guarantee a positive operator with row sum equal to 1. Similarly, Cohen et al. [16] restrict the minimization problem to operators that are symmetric. Multiple other publications deal with constraints that enforce a low-rank formulation of DMD. A closed-form solution with optimal  $\ell_2$ -norm is provided by Héas and Herzet [27], who also give an overview of sub-optimal approaches [26]. Although our method constrains the DMD-specific minimization problem in principle, we directly address the spectral theoretic properties (i.e., eigenvalues and thus frequencies), whereas the approaches mentioned above address algebraic properties.

To the best of our knowledge, the publication of Krake et al. [33] is the only paper that directly integrates frequency-based constraints into DMD. They provide a DMD-based method for video foreground/background segmentation that computes a static background model. For this step, a single constraint (DMD must contain the eigenvalue  $\lambda = 1$ ) is integrated into DMD. While our approach uses a partly similar derivation, we end up with a formulation that allows integrating multiple different eigenvalues (frequencies) into DMD. In addition to that, our formulation also enables a human-in-the-loop process.

Besides DMD-based approaches, the most common techniques for the visualization and analysis of time series are seasonal trend decomposition using LOESS (STL) [15] and singular spectrum analysis (SSA). STL decomposes a univariate time series into three components: trend, season, and residual. It is used for visual analysis, identification of anomalies [13, 28], and forecasting [39]. In contrast, SSA is a spectral decomposition that produces multiple oscillatory modes characterizing specific frequencies (although SSA is not connected to a frequency domain representation). Due to the generic formulation, SSA has various applications [2, 4, 22, 25, 31, 40]. A more recent development is the use of an HJ-biplot for an explorative visualization [17]. Besides the fact that DMD produces conceptually a different decomposition, we incorporate multiple frequency-based constraints into DMD.

The approaches mentioned before only partially take advantage of user interaction. More interaction-oriented techniques combine representations such as a calendar-based view [55], spiral view [38, 57], time wheel [51], or parallel coordinates [24] with tools like autocorrelation, bias reduction, aggregation, and discrete Fourier transform to analyze [12, 30, 56, 58], segment [3, 42], filter [7], or predict [5]

time series. There are also direct extensions of previously mentioned techniques that integrate interaction. Carlis et al. [11] and Tominski and Schumann [52] analyze time series for the identification of seasonal and cyclic patterns via an interactive spiral view. Another related topic is the interaction with machine learning via visualizations [29, 44]. All these publications address goals that are similar to ours. However, our explorative visualization is based on a spectral decomposition, which conceptually differs a lot. In addition, our user interaction facilitates the incorporation of frequencies, which is not covered in the above works.

### 3 PROBLEM STATEMENT

In this section, our goal is to discuss which problems we want to address and how they are typically solved using the proposed human-in-the-loop process. Furthermore, we specify the requirements for DMD.

#### 3.1 Introductory Example

In the following, we want to describe how a user approaches certain problems with our method. This will also emphasize which types of problems can be addressed. We consider the same time series as in Fig. 1, i.e., the task is to analyze daily data. This is the first information the user is confronted with, as illustrated in Fig. 2. However, the user has no influence on the application of conventional DMD to data. The computation results in DMD components that highlight a trend and two seasonal patterns with the periods 18.24 and 6.90, respectively.

In general, the user has to accept the patterns and can use post-processing techniques to enhance the results. However, they do not influence the core mechanism of DMD. Our approach enables incorporating multiple frequency-based constraints into DMD. The corresponding algorithm is designed such that this process can be performed by a feedback-loop. This means, for instance, that the user may observe an interesting pattern, such as the period 6.90, and wants to adjust it. Furthermore, the user can connect observed patterns with external knowledge (here, the connection to weekly patterns). This is the first user-specified update in Fig. 2: the adjustment of the period 6.90.

The updated DMD components still produce a pattern (period 17.05) that does not match with daily data. Therefore, the user may try to integrate another frequency that has a relation to the data. Here, the user tries to integrate a two-weekly pattern, i.e., the period 14 into the system by trial and error. It can be observed that the updated DMD components replace the period 17.05 by the period 14.00. However, this increases the error significantly. Therefore, the user decides to decline this change and, instead, integrates the next reasonable choice of frequency: a monthly pattern. The final result seems to be reasonable because the decomposition exhibits relevant time scale features in the data.

#### 3.2 Requirements for DMD

In the previous subsection, we described a typical situation for our approach. It shows that our technique has to be computationally efficient to facilitate human-in-the-loop interaction for the user. This aspect must be taken into account for the mathematical modeling. Moreover, due to limited space and a typically large number of DMD components, a filtering is necessary to highlight the relevant components. In addition to that, the user should keep track of changes in case of multiple updates (in particular, if updates are performed by trial and error).

**Algorithm 1:** Constrained Dynamic Mode Decomposition

---

**Input** : data  $x_0, \dots, x_m \in \mathbb{R}^n$  (with time delay embedding)  
**Output** : DMD eigenvalues  $\lambda_j$ , modes  $\vartheta_j$ , and amplitudes  $a_j$

- 1  $X \leftarrow [x_0 \mid \dots \mid x_{m-1}] \in \mathbb{R}^{n \times m}$ ;
- 2  $Y \leftarrow [x_1 \mid \dots \mid x_m] \in \mathbb{R}^{n \times m}$ ;
- 3  $[U, \Sigma, V] \leftarrow \text{Thin.SVD}(X)$  with  $\text{rank}(X) = m$ ;
- 4  $S \leftarrow U^* Y V \Sigma^{-1} \in \mathbb{R}^{m \times m}$ ; // Equation 5
- 5  $[\Lambda, W] \leftarrow \text{EIG}(S)$ ; // eigenvalue solver
- 6  $\Theta = [\vartheta_1 \mid \dots \mid \vartheta_m] \leftarrow Y V \Sigma^{-1} W \Lambda^{-1}$ ;
- 7  $a = (a_1, \dots, a_m) \leftarrow \text{LS}(\Theta \Lambda, x_1)$ ; // least-squares solver
- 8 **while** (DMD is inappropriate) **do**
- 9     define frequencies  $\tilde{\lambda}_1, \dots, \tilde{\lambda}_q$ ; // human-in-the-loop
- 10      $K \leftarrow \text{diag}(K_1, \dots, K_m)$ ; // Equation 20
- 11      $\tilde{\mathcal{R}} \leftarrow (G^* K^* K G)$ ;
- 12      $\tilde{\mathcal{R}}^{-1} \tilde{G}^* \leftarrow \text{LS}(\tilde{\mathcal{R}}, \tilde{G}^*)$ ; // least-squares solver
- 13      $c \leftarrow V \Sigma^{-1} U^* x_m$ ;
- 14      $\tilde{c} \leftarrow c + \tilde{\mathcal{R}}^{-1} \tilde{G}^* (\tilde{G} \tilde{\mathcal{R}}^{-1} \tilde{G}^*)^{-1} (\tilde{b} - \tilde{G} c)$ ; // Equation 17
- 15      $\tilde{S} \leftarrow (\Sigma V^*) \tilde{C} (V \Sigma^{-1})$ ; // Equation 18
- 16      $[\tilde{\Lambda}, \tilde{W}] \leftarrow \text{EIG}(\tilde{S})$ ; // eigenvalue solver
- 17      $\tilde{\Theta} = [\tilde{\vartheta}_1 \mid \dots \mid \tilde{\vartheta}_m] \leftarrow Y V \Sigma^{-1} \tilde{W} \tilde{\Lambda}^{-1}$ ;
- 18      $\tilde{a} = (\tilde{a}_1, \dots, \tilde{a}_m) \leftarrow \text{LS}(\tilde{\Theta} \tilde{\Lambda}, x_1)$ ; // least-squares solver

---

## 4 MATHEMATICAL BACKGROUND

In this section, we will shortly explain the theory of delay embedding and how we use it in the context of DMD. In addition, the theoretical and algorithmic concept of DMD is presented.

### 4.1 Time Series Data and Delay Embedding

Although our approach is applicable to any kind of data, we describe it for real-valued time series. Given a time series  $z_0, \dots, z_M \in \mathbb{R}^N$ , the number of attributes is  $N$  and the number of time steps is  $M+1$ . In general, for high-dimensional data (as from many examples of simulation-based or experimental data), the relation  $N \gg M$  holds (leading to linearly independent vectors), which is the typical condition for DMD. In this case, DMD can directly be applied to the time series  $z_0, \dots, z_M$  without any further processing. In contrast, for univariate time series or multivariate time series with only a few attributes, the resulting relation  $N \ll M$  usually introduces large errors in the reconstruction by DMD and an inaccurate identification of frequencies. A commonly used technique to avoid this DMD-specific dimensionality problem is time delay embedding, which is also called hankelization [18, 19, 21, 32, 48, 50].

Developed in the field of dynamical systems, time delay embedding addresses the problem of reconstructing (chaotic) dynamics in state space with a sequence of observations [49]. In our discrete setting, the procedure of time delay embedding corresponds to lifting the time series  $z_0, \dots, z_M \in \mathbb{R}^N$  into a higher-dimensional space  $\mathbb{R}^n$  by stacking  $d$  time-shifted copies. We describe this via  $\mathcal{H}_d: \mathbb{C}^{N \times M+1} \rightarrow \mathbb{C}^{n \times m+1}$ ,

$$[z_0 \mid \dots \mid z_M] \mapsto \begin{pmatrix} z_0 & z_1 & \dots & z_{M-d} \\ z_1 & z_2 & \dots & z_{M-d+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_d & z_{d+1} & \dots & z_M \end{pmatrix}, \quad (1)$$

where we have the relations  $n = N(d+1)$  and  $m = M-d$  with the delay parameter  $d \geq 0$ . The delayed time series is a hankel-type matrix (in general, not a square matrix). Each row corresponds to a windowed representation of the original time series. The delayed time series solves the DMD-specific problem of dimensionality because for real world data the delay parameter  $d$  can be chosen such that the columns of the hankel-type matrix are linearly independent. In this context, the relation  $n \geq m+1$  must be satisfied. Inserting the variables  $n$  and  $m$ , this leads to the condition  $d \geq (M-N+1)/(N+1)$  for the delay parameter  $d$ . For univariate time series ( $N=1$ ), the condition

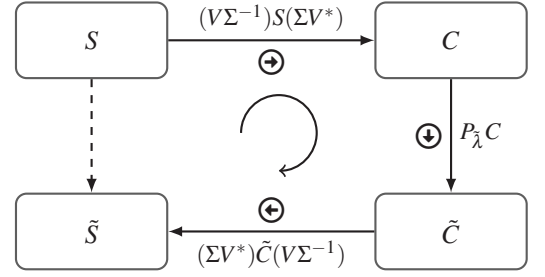


Fig. 3. An overview of our mathematical derivation: the commutative diagram shows the process of computing the modified matrix  $\tilde{S}$ .

boils down to  $d \geq M/2$ . However, we recommend selecting a delay parameter that is slightly larger than  $M/2$ . In our evaluations,  $d$  was chosen as approximately  $d = 0.6 \cdot M$ . For these values, our method performs in a reasonable and stable way. Choosing values near  $M/2$  or  $M$  can lead to insufficiencies. This aspect is shown in a parameter study in the supplemental material [34].

Given objects in the delayed space  $x_0, \dots, x_m \in \mathbb{R}^{n \times m+1}$  with the relations  $n = N(d+1)$  and  $m = M-d$ , the procedure of reverting the higher-dimensional embedded objects to the original space is crucial (especially for our approach). The common technique is to use diagonal averaging (actually, the anti-diagonals are averaged). We describe this operation with  $\mathcal{H}_d^+: \mathbb{C}^{n \times m+1} \rightarrow \mathbb{C}^{N \times M+1}$ ,

$$\begin{pmatrix} x_{0,0} & x_{1,0} & \dots & x_{M-d,0} \\ x_{0,1} & x_{1,1} & \dots & x_{M-d,1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{0,d} & x_{1,d} & \dots & x_{M-d,d} \end{pmatrix} \mapsto \left[ x_{0,0} \mid \frac{x_{0,1}+x_{1,0}}{2} \mid \frac{x_{0,2}+x_{1,1}+x_{2,0}}{3} \mid \dots \right]. \quad (2)$$

It can be observed that diagonal averaging is a left inverse operation for time delay embedding, i.e., the following identity holds:  $\mathcal{H}_d^+(\mathcal{H}_d[z_0 \mid \dots \mid z_M]) = [z_0 \mid \dots \mid z_M]$ . Hence, the interaction with DMD is as follows: DMD is applied to the delayed time series (to overcome the dimensionality problem mentioned before) and diagonal averaging is used to relate our DMD results to the original space.

### 4.2 Dynamic Mode Decomposition

DMD is a spectral decomposition technique that extracts spatio-temporal patterns from data. As mentioned before, we apply DMD to time-delayed data  $[x_0 \mid \dots \mid x_m] = \mathcal{H}_d[z_0 \mid \dots \mid z_M] \in \mathbb{R}^{n \times m+1}$  to overcome the DMD specific issue of dimensionality. Hence, we can assume the relation  $n > m+1$  and the linear independence of the vectors. Before we explain the algorithm of DMD shown in the first part of Algorithm 1 (lines 1–7), we describe the principles of DMD [36, 37], which help understand our approach in the next section.

In general, DMD efficiently computes eigenvalues and eigenvectors of a high-dimensional matrix  $A \in \mathbb{R}^{n \times n}$ , which determine the frequency-based decomposition of the data, via a low-dimensional representation  $S$ . The matrix  $A$  is characterized by the minimization problem

$$\min \|AX - Y\|_F^2, \quad (3)$$

where  $X = [x_0 \mid \dots \mid x_{m-1}]$  and  $Y = [x_1 \mid \dots \mid x_m]$ . Since this minimization problem usually does not have a unique solution (due to the fact  $n > m$ ), DMD uses the minimum-norm solution that is given by

$$A = YX^+ \in \mathbb{R}^{n \times n}, \quad (4)$$

where  $X^+$  is the Moore-Penrose pseudoinverse of  $X$ . To avoid computationally costly operations for the spectral decomposition of  $A$ , the low-dimensional representation  $S \in \mathbb{R}^{r \times r}$  of  $A$  is computed via the thin singular value decomposition (SVD) of  $X$ , where irrelevant rows and columns in the square matrix representation are discarded, i.e.,  $X = U \Sigma V^*$  with  $U \in \mathbb{R}^{n \times r}$ ,  $\Sigma \in \mathbb{R}^{r \times r}$ ,  $V \in \mathbb{R}^{m \times r}$ , and  $r = \text{rank}(X) = m$ :

$$S = U^* A U = U^* Y V \Sigma^{-1} U^* U = U^* Y V \Sigma^{-1} \in \mathbb{R}^{r \times r}. \quad (5)$$

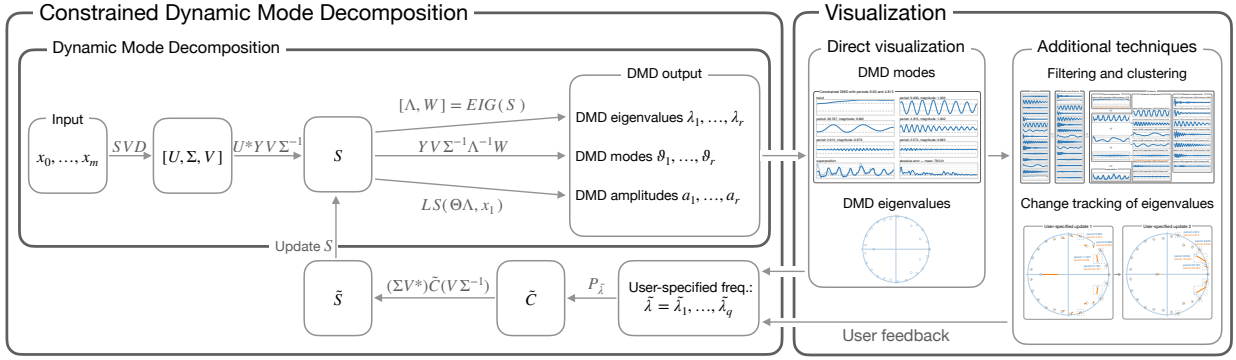


Fig. 4. A flowchart-like overview of our approach. After the application of DMD, the DMD output is processed in the direct visualization, which may give initial insights into DMD. On this basis, the user can decide to consult the additional visualization techniques including clustering and change tracking of eigenvalues or to incorporate desired frequencies via constrained DMD.

This low-dimensional matrix is the key to efficiently compute the spectral components of  $A$  that characterize the data spatio-temporally.

An algorithmic description is provided in the first part of Algorithm 1 (lines 1–7). Starting with the definition of  $X$  and  $Y$  (lines 1–2), the thin SVD of  $X$  (line 3) is used to compute the low-dimensional matrix  $S \in \mathbb{R}^{m \times m}$  (line 4) via Equation 5. The next step is to compute the non-zero eigenvalues  $\lambda_j \in \mathbb{C}$  and eigenvectors  $v_j \in \mathbb{C}^m$  of  $S$  (line 5). Then, the eigenvectors are transformed into DMD modes  $\vartheta_j = \frac{1}{\lambda_j} YV\Sigma^{-1}v_j \in \mathbb{C}^n$  ( $\vartheta_j$  is an eigenvector of the matrix  $A$  with eigenvalue  $\lambda_j$  [36]) and are arranged column-wise in the matrix  $\Theta$  (line 6). Finally, the DMD amplitudes  $a_j \in \mathbb{C}$  are computed by solving  $\min \|\Theta\Lambda a - x_1\|$  via a least-squares solver (line 7). In general, this algorithm leads to an approximate decomposition of the data. Krake et al. [36] proved that this representation is exact if the matrices  $X$  and  $Y$  are full-rank and the eigenvalues  $\lambda_1, \dots, \lambda_m$  are distinct (in this case, it holds  $r = m$ ):

$$x_k = \sum_{j=0}^m \lambda_j^k a_j \vartheta_j, \quad (6)$$

where  $\lambda_0 = 0$ ,  $a_0 = -\sum_j \frac{1}{\lambda_j} \prod_{k \neq j} \frac{1}{\lambda_j - \lambda_k}$ , and  $\vartheta_0 = x_m - XX^+x_m$  (these are only used for the reconstruction of  $x_0$  and do not contribute to the dynamics). The equation provides an interpretation of the components  $\lambda_j$ ,  $\vartheta_j$ , and  $a_j$ , which is important to understand our approach.

## 5 METHOD

This section presents our method: constrained DMD. We start with a motivation that illustrates the potential of DMD. Afterward, the formulation for incorporating multiple frequency-based constraints is derived and the corresponding human-in-the-loop algorithm is provided, which is publicly available<sup>1</sup>. We conclude the section with additional visualization techniques that complement the visual exploration.

### 5.1 Motivation

DMD decomposes data into three types: eigenvalues, modes, and amplitudes. According to Equation 6, the time-delayed temporal evolution of the  $j$ -th DMD components is given by  $\bar{x}_k(j) = \lambda_j^k a_j \vartheta_j \in \mathbb{C}^n$  for  $k = 0, \dots, m$ . To relate it to the original space (such that specific patterns in the data are characterized), diagonal averaging needs to be applied:  $[\bar{z}_0(j) \mid \dots \mid \bar{z}_m(j)] = \mathcal{H}_d^+([\bar{x}_0(j) \mid \dots \mid \bar{x}_m(j)]) \in \mathbb{C}^{N \times M+1}$ . In the case of univariate data, the application of DMD and constrained DMD is demonstrated in Fig. 1. The time series used in this figure (middle box) is a typical example of daily data. Besides a linear trend and a noise, two seasonal patterns (periods 7 and 28) are part of the superposition, which correspond to weekly and monthly patterns, respectively.

The application of both DMDs leads to many components, where the three most dominant ones are depicted in the figure (sorting top to

bottom). The remaining components characterize noise patterns. While conventional DMD (left box) nearly identifies the weekly pattern with the period 6.90, it does not correctly represent the linear trend as well as the other seasonal pattern. In fact, DMD assigns the period 18.24 to the monthly pattern. Due to the natural spatio-temporal representation of DMD (assignment of spatial components to a frequency and growth rate), an incorporation of frequency-based constraints is feasible, which is realized by our constrained DMD (right box). This interaction can be based on a human-in-the-loop process (e.g., inspired by previous results of DMD) or be motivated by external aspects. Since the time series consists of daily data, the time series will exhibit weekly and monthly patterns. Choosing these patterns as constraints, our constrained DMD produces analytically correct periods. The corresponding temporal evolutions of the components identify the patterns of the time series correctly. This shows the potential of our approach.

### 5.2 Mathematical Derivation

As motivated in the previous subsection, our goal is to incorporate frequency-based constraints into DMD to enable interaction with the core algorithm. To realize this, we first recap that the linearly independent time-delayed data  $[x_0 \mid \dots \mid x_m] = \mathcal{H}_d[z_0 \mid \dots \mid z_M] \in \mathbb{R}^{n \times m+1}$  with the respective data matrices  $X$  and  $Y$  as well as the reduced singular value decomposition of  $X = U\Sigma V^*$  result in the low-dimensional matrix  $S = U^*YV\Sigma^{-1} \in \mathbb{R}^{m \times m}$  (see Equation 5 or lines 1–4 in Algorithm 1). This is the start point of our derivation and highlighted in the top left of Fig. 3. The figure shows an overview of the entire derivation in the form of a commutative diagram. The dashed arrow indicates the goal, that is to transform the matrix  $S$  such that the modified matrix  $\tilde{S}$  has desired eigenvalues (which we demand), a reasonable connection to DMD, and numerical advantages for an efficient computation. In the following, we will derive a formulation for the matrix  $\tilde{S}$  following the commutative structure in Fig. 3 with steps  $\oplus$ ,  $\ominus$ , and  $\ominus$ .

**Derivation step  $\oplus$**  Another way of formulating DMD is to use a companion matrix  $C$ . To facilitate the next derivation steps, we recap that a companion matrix (according to our definition) has the structure

$$C = \begin{pmatrix} 0 & \dots & 0 & c_0 \\ 1 & \dots & 0 & c_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & c_{m-1} \end{pmatrix} \in \mathbb{R}^{m \times m}. \quad (7)$$

In this case, instead of solving the minimization problem (Equation 3), the related minimization problem  $\min \|XC - Y\|_F$  is considered, which boils down to  $\min \|Xc - x_m\|_2$ . A solution is given by

$$c = X^+x_m. \quad (8)$$

The relation between the two concepts is given by [36, Lemma 5.2]

$$C = (V\Sigma^{-1})S(V\Sigma^*), \quad (9)$$

<sup>1</sup><https://github.com/kraketm/ConstrainedDMD>

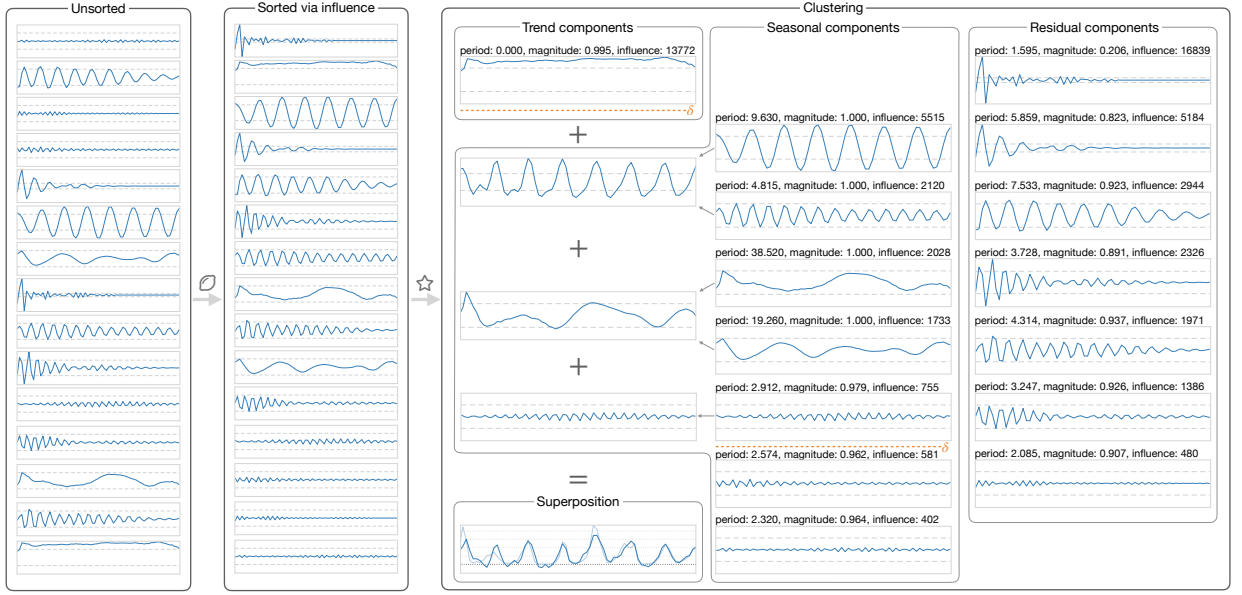


Fig. 5. An illustration of our filtering and clustering. The unstructured temporal evolutions are sorted in step  $\bigcirc$  via their influence. Then, our filtering is applied in step  $\star$ , resulting in a grouped representation (trend, seasonal, and residual components). Finally, the components are clustered, where properties such as harmonic patterns are considered. The resulting superposition (dark blue) is compared to the original time series (light blue).

which demonstrates that the two matrices are similar, i.e., they are equal up to a change of basis. Hence, they have the same eigenvalues.

**Derivation step  $\oplus$**  Whereas the matrix  $S$  has a more compact numerical representation (due to the SVD), the companion matrix  $C$  has direct relation to the eigenvalues. In fact, the characteristic polynomial of  $C$  is determined by

$$p_C(\lambda) = -c_0 - c_1\lambda - \dots - c_{m-1}\lambda^{m-1} + \lambda^m. \quad (10)$$

This connection enables a modification of the eigenvalues. (Although Krake et al. [33] also use this connection to enforce the eigenvalue  $\lambda = 1$ , they do not exploit the similarity of  $S$  and  $C$ , leading to a different formulation). In fact,  $\lambda_j$  is an eigenvalue of  $C$  if and only if  $p_C(\lambda_j) = 0$ . Since the matrix  $C$  (or the similar matrix  $S$ ) has the  $m$  eigenvalues  $\lambda_1, \dots, \lambda_m$ , the vector  $c$  and therefore the companion matrix  $C$  is fully characterized by the following system of linear equations:

$$Gc = b, \quad (11)$$

where  $G$ , a Vandermonde matrix, and  $b$  are given by

$$G = \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^{m-1} \\ 1 & \lambda_2 & \dots & \lambda_2^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_m & \dots & \lambda_m^{m-1} \end{pmatrix} \in \mathbb{C}^{m \times m}, \quad b = \begin{pmatrix} \lambda_1^m \\ \lambda_2^m \\ \vdots \\ \lambda_m^m \end{pmatrix} \in \mathbb{C}^m. \quad (12)$$

We use this representation to formulate an analogous (underdetermined) system encoding our desired eigenvalues  $\tilde{\lambda}_1, \dots, \tilde{\lambda}_q$  with  $q \leq m$ :

$$\tilde{G} = \begin{pmatrix} 1 & \tilde{\lambda}_1 & \dots & \tilde{\lambda}_1^{m-1} \\ 1 & \tilde{\lambda}_2 & \dots & \tilde{\lambda}_2^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \tilde{\lambda}_q & \dots & \tilde{\lambda}_q^{m-1} \end{pmatrix} \in \mathbb{C}^{q \times m}, \quad \tilde{b} = \begin{pmatrix} \tilde{\lambda}_1^m \\ \tilde{\lambda}_2^m \\ \vdots \\ \tilde{\lambda}_q^m \end{pmatrix} \in \mathbb{C}^q. \quad (13)$$

The key is now to determine how these two systems can be merged in a meaningful and numerically efficient way. The canonical method is to formulate a minimization problem that integrates the desired frequencies as a constraint:

$$\min_{\tilde{c}} \|G\tilde{c} - b\|_2^2 \quad \text{subject to } \tilde{G}\tilde{c} = \tilde{b} \quad (14)$$

Due to Equation 11, we know that  $c$  is a solution to the minimization problem without constraints and, according to the appendix, a solution is therefore given by the formula

$$\tilde{c}_{bad} = c + \mathfrak{G}^{-1}\tilde{G}^* \left( \tilde{G}\mathfrak{G}^{-1}\tilde{G}^* \right)^{-1} (\tilde{b} - \tilde{G}c), \quad (15)$$

where we define  $\mathfrak{G}^{-1} = (G^*G)^{-1}$ . Although this solution encodes the desired eigenvalues (implicitly in the vector  $c$ ), it can radically alter the identification of other eigenvalues (which we do not address by our desired eigenvalues). This can also be observed in the weighting matrix  $\mathfrak{G}$  in Equation 15, where only  $G$  is encoded without any information about the importance of certain eigenvalues. For this reason, we suggest integrating a scaling matrix into the minimization problem. Let  $K \in \mathbb{R}^{m \times m}$  be a diagonal matrix that describes the importance of eigenvalues (a reasonable choice will be discussed in the next subsection), then, the improved minimization problem is given by

$$\min_{\tilde{c}} \|K(G\tilde{c} - b)\|_2^2 \quad \text{subject to } \tilde{G}\tilde{c} = \tilde{b} \quad (16)$$

Again, according to the appendix, a solution to this problem is given by

$$\tilde{c} = P_{\tilde{\lambda}}(c) = c + \mathfrak{K}^{-1}\tilde{G}^* \left( \tilde{G}\mathfrak{K}^{-1}\tilde{G}^* \right)^{-1} (\tilde{b} - \tilde{G}c), \quad (17)$$

where we define  $\mathfrak{K}^{-1} = (G^*K^*KG)^{-1}$ . In fact, if the scaling matrix  $K$  is the identity matrix, we get the same solution as for Equation 15. However, for a meaningful choice of  $K$ , we obtain a solution that both encodes the desired eigenvalues and does minimal changes to DMD. This effect can also be observed in the weighting matrix  $\mathfrak{K}$ .

**Derivation step  $\oplus$**  The result of the previous step is a new companion matrix  $\tilde{C}$  that has the desired eigenvalues via the vector  $\tilde{c}$  implicitly encoded. Finally, the companion matrix  $\tilde{C}$  needs to be retransformed to be aligned with the original DMD algorithm. According to Equation 9, the retransformation is given by

$$\tilde{S} = (\Sigma V^*)\tilde{C}(V\Sigma^{-1}). \quad (18)$$

This concludes our derivation of the modified matrix  $\tilde{S}$ .

### 5.3 Algorithm and Human-in-the-loop

Now, we describe the implementation and discuss the human-in-the-loop process. Figure 4 provides an overview of the process. The inner box shows the conventional DMD approach, which is described in lines 1–7 in Algorithm 1. The resulting DMD output is then used for further processing and visualizations. A natural step is to directly visualize the DMD eigenvalues, modes, and amplitudes first. There are different techniques described in the literature, however, in this paper we use more standard visualizations to clearly demonstrate our approach (our goal is not to improve the direct visualizations of DMD components):

- DMD eigenvalues are visualized in the complex plane (later on, this representation is used for the tracking of eigenvalues).
- DMD modes and amplitudes are represented by their temporal evolutions  $[\bar{z}_0(j) \mid \dots \mid \bar{z}_m(j)] = \mathcal{H}_d^+([\bar{x}_0(j) \mid \dots \mid \bar{x}_m(j)]) \in \mathbb{C}^{N \times M+1}$ , where  $\bar{x}_k(j) = \lambda_j^k a_j \vartheta_j \in \mathbb{C}^n$  for  $k = 0, \dots, m$ .

Besides the visualization, a sorting of DMD components is important as well due to the large number of components (see Fig. 5 step  $\mathcal{D}$ ). We recommend sorting the components by their influence onto the time series. The influence of components is not only important for the representation, it is also crucial for our constrained DMD algorithm. A possible definition of the influence will be derived in the following.

First of all, we need to emphasize that eigenvalues  $\lambda_j$ , scaled modes  $a_j \vartheta_j$ , and therefore the temporal evolutions  $\lambda_j a_j \vartheta_j$  occur in complex conjugate pairs in the time-delayed space [35]. Thus, complex conjugated pairs can be combined as follows: Assuming  $j_1$  and  $j_2$  to be complex conjugate pairs, then the temporal evolution is given by

$$\begin{aligned} & [\bar{z}_0(j_1) \mid \dots \mid \bar{z}_m(j_1)] + [\bar{z}_0(j_2) \mid \dots \mid \bar{z}_m(j_2)] \\ &= \mathcal{H}_d^+[\bar{x}_0(j_1) \mid \dots \mid \bar{x}_m(j_1)] + \mathcal{H}_d^+[\bar{x}_0(j_2) \mid \dots \mid \bar{x}_m(j_2)] \\ &= \mathcal{H}_d^+[\bar{x}_0(j_1) + \bar{x}_0(j_2) \mid \dots \mid \bar{x}_m(j_1) + \bar{x}_m(j_2)] \\ &= \mathcal{H}_d^+[2 \cdot \Re(\lambda_{j_1}^0 a_{j_1} \vartheta_{j_1}) \mid \dots \mid 2 \cdot \Re(\lambda_{j_1}^m a_{j_1} \vartheta_{j_1})], \end{aligned} \quad (19)$$

where we used the following relation  $\lambda_{j_1}^k a_{j_1} \vartheta_{j_1} + \lambda_{j_2}^k a_{j_2} \vartheta_{j_2} = \lambda_{j_1}^k a_{j_1} \vartheta_{j_1} + \text{conj}(\lambda_{j_1}^k a_{j_1} \vartheta_{j_1}) = 2\Re(\lambda_{j_1}^k a_{j_1} \vartheta_{j_1})$ . This shows that the temporal evolution of a DMD component is actually given by its real part and, therefore, we recommend not showing temporal evolution twice. The influence of a DMD component can be computed by (multivariate data with various scales must be additionally aligned)

$$K_j = \left\| \mathcal{H}_d^+[\Re(\lambda_{j_1}^0 a_{j_1} \vartheta_{j_1}) \mid \dots \mid \Re(\lambda_{j_1}^m a_{j_1} \vartheta_{j_1})] \right\|_F. \quad (20)$$

This scalar is not only an appropriate value for the sorting of DMD components (compare Fig. 5), but also a suitable choice for the scaling matrix  $K = \text{diag}(K_1, \dots, K_m)$  for constrained DMD. Computing these scalars is also the first step of our constrained DMD if the feedback-loop is considered in Fig. 4. As we will see later, the user feedback is useful if the conventional DMD detects frequencies only approximately (see Fig. 1) or if the visualization suggests certain frequencies. It is important that frequencies are incorporated via complex conjugated pairs, i.e., the incorporation of a period  $p \in \mathbb{R}$  is given by

$$p \longrightarrow \tilde{\lambda}_1 = e^{2\pi i/p} \quad \text{and} \quad \tilde{\lambda}_2 = e^{-2\pi i/p}. \quad (21)$$

The human-in-the-loop process is algorithmically described in more detail in Algorithm 1 (lines 8–18). It is formalized via a while loop that stops if the user considers DMD to be appropriate. The definition of user-specified frequencies  $\tilde{\lambda}_1, \dots, \tilde{\lambda}_q$  is the first step (line 9). Then, the scaling matrix  $K$  and the weighting matrix  $\mathfrak{R}$  are computed (lines 10–11). Afterward, the term  $\mathfrak{R}^{-1} \tilde{G}^*$  is computed via solving  $q$  systems of linear equations (line 12). This result as well as the computed vector  $c$  (line 13) is used to obtain the modified vector  $\tilde{c}$  via Equation 17 (line 14). With this, we can compute the modified matrix  $\tilde{S}$  (line 15). As indicated in the overview, Fig. 4, the modified matrix  $\tilde{S}$  can be seen as an update of the matrix  $S$ . This procedure is described in the remaining steps of Algorithm 1, which are simply a repetition of lines 5–7.

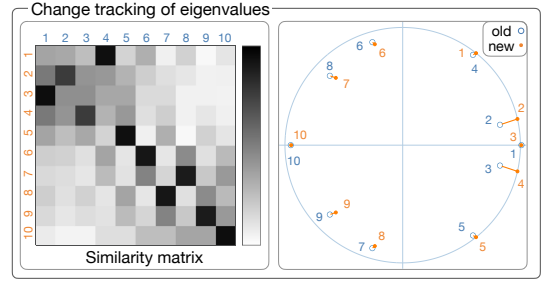


Fig. 6. Illustration of the change tracking of eigenvalues. A specific DMD-based similarity matrix is used for the identification of pairs. For the visualization, the complex plane and a vector-like connectivity are used.

Algorithm 1 shows that our human-in-the-loop process exploits previous DMD computations (e.g., the SVD of  $X$  is only performed once). Furthermore, we emphasize that only the objects with a tilde symbol need to be updated, i.e., lines 10, 11, and 13 need to be computed once. In addition, the computation of the modified matrix  $\tilde{S}$  does not produce much overhead, as the inverses can be computed efficiently: In fact, line 12 uses a solver in the low-dimensional space  $m$  with time complexity  $\mathcal{O}(m^3)$  and line 14 computes the inverse of a  $q \times q$  matrix with a time complexity of  $\mathcal{O}(q^3)$ . Since  $q$  is the number of frequency-based constraints, it satisfies  $q \leq m$ , however, it is much smaller in general. Thus, the update of  $S$  computed in lines 9–15 does not produce much overhead. This applies to the eigenvalue computation in line 16 as well because the time complexity is  $\mathcal{O}(m^3)$ . In the last two lines 17–18, partly high-dimensional matrices are involved in the computation such that the time complexity is given by  $\mathcal{O}(nm^2)$ . In sum, constrained DMD has a time complexity of  $\mathcal{O}(nm^2)$ , however, most of the computations are performed in a much lower-dimensional space. In addition to that, all solvers (EIG, LS, or Thin\_SVD) can be parallelized [6, 8, 47].

### 5.4 Additional Visualization Techniques

In the previous section, an overview of the human-in-the-loop process was presented. The feedback-loop can be performed after the first impression of the direct visualization or as a consequence of our additional visualization techniques. These are presented in the following.

**Filtering and clustering** In the previous subsection, we discussed the direct visualization of DMD components (compare Fig. 4). The eigenvalues are represented in the complex plane and the modes are visualized by the corresponding temporal evolution (of course, more advanced techniques can be used in the case of multivariate data). As proposed previously, we recommend a sorting via their influence to suppress misleading DMD components (see Fig. 5, step  $\mathcal{D}$ ). While this sorting is useful for a first impression, it is not optimal for the task of clustering components. Thus, we propose a filtering that takes the location of the eigenvalues in the complex plane into account.

In general, the location of an eigenvalue  $\lambda_j$  in the complex plane determines the dynamics of the temporal evolution  $\lambda_j^k a_j \vartheta_j$ . Eigenvalues on the positive real axis, i.e., the imaginary part  $\Im(\lambda_j)$  is small and  $\Re(\lambda)$  is positive, characterize (long-term) increase or decrease in the data and therefore correspond to the trend. In contrast, eigenvalues close to the unit circle, i.e.,  $|\lambda| \approx 1$ , (which are not classified as trend components) describe oscillatory structures and therefore seasonal patterns. The remaining components characterize noise. We can use this information to group the DMD components into the following subsets:

$$\mathcal{M}_T = \{j \in \{1, \dots, m\} : |\Im(\lambda_j)| < \varepsilon_T \text{ and } \Re(\lambda_j) > 0\}, \quad (22)$$

$$\mathcal{M}_S = \{j \in \{1, \dots, m\} : ||\lambda_j| - 1| < \varepsilon_S\} \setminus \mathcal{M}_T, \quad (23)$$

$$\mathcal{M}_R = \{1, \dots, m\} \setminus (\mathcal{M}_T \cup \mathcal{M}_S), \quad (24)$$

where  $\varepsilon_T$  and  $\varepsilon_S$  are user-defined parameters. Thus, we can create an overview representation (note that temporal evolutions occur in complex conjugate pairs). We use the disjoint union of the index set

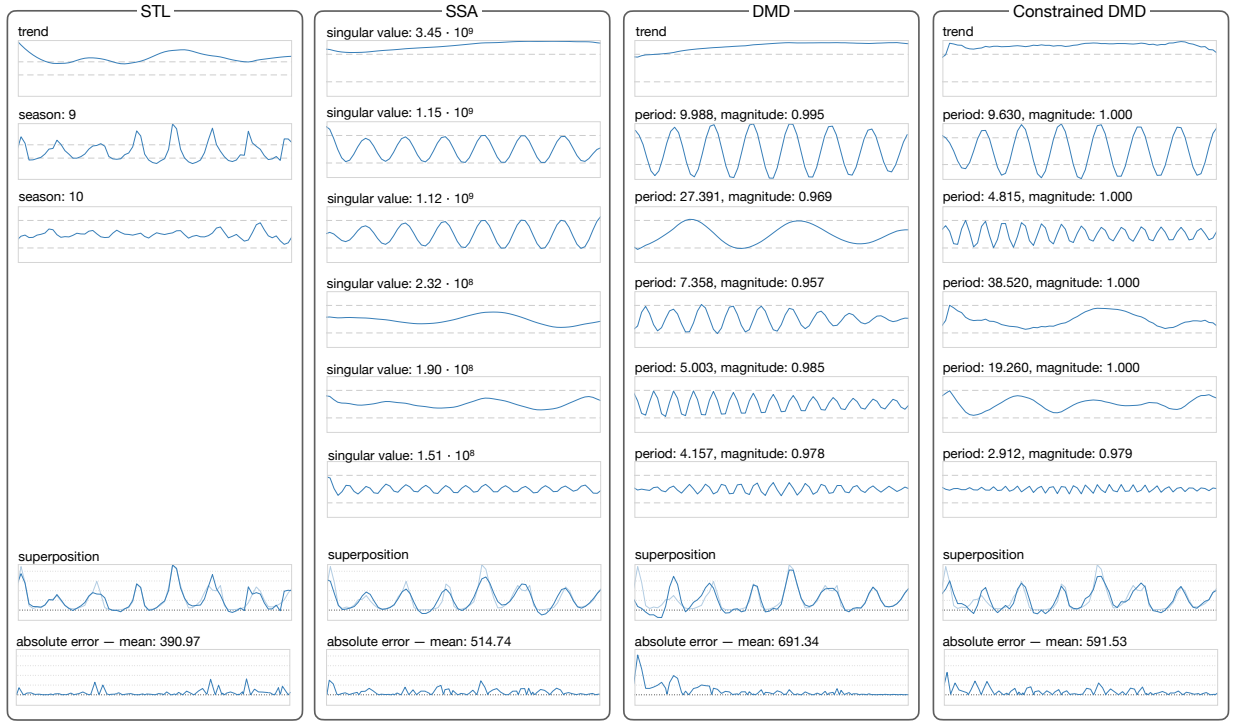


Fig. 7. Comparison of different spatio-temporal decompositions: STL, SSA, DMD, and constrained DMD. The methods are applied to the lynx dataset that consists of cyclic and seasonal patterns along with a slightly increasing trend. According to animal populations in North Canada, the population is periodic with the period 9.63. Constrained DMD incorporates this period and harmonic frequencies. The resulting components of each method are visualized, where the dashed line indicate the values 1000 and -1000. Besides the components, the superposition of these are presented and the absolute error. For these plots, the dotted lines correspond to the increase in steps of 1500, where the bold dotted line indicates the value 0.

$\{1, \dots, m\}$  via  $\mathcal{M}_T$ ,  $\mathcal{M}_S$ , and  $\mathcal{M}_R$  and sort inside of these categories regarding the influence of the temporal evolutions (via  $K_j$ , see Equation 20). A visualization of this filtering is shown in Fig. 5 in step  $\star$ .

The next step is to use the filtering for clustering. We propose only clustering components that satisfy a threshold  $\delta$ . The clustering of trend components can be done by the superposition of these components, i.e.,  $\mathcal{C}_T^\delta = \{j \in \mathcal{M}_T : K_j \geq \delta\}$ . In contrast, seasonal components should be aggregated into different clusters  $\mathcal{C}_{S_1}^\delta, \mathcal{C}_{S_2}^\delta, \dots$  that consist of harmonics. This step must be done in a reasonable way to highlight certain patterns in the data. In this regard, constrained DMD has enormous advantages due to incorporating frequencies (later on, we purposely choose harmonic frequencies). An illustration of the clustering process is given in Fig. 5. The residual components  $\mathcal{C}_R$  complement the representation (and help in the case of wrong assignments).

**Change tracking** Due to our human-in-the-loop process shown in Fig. 4, constrained DMD incorporates new frequencies. This process affects all components of the DMD decomposition. Although our mathematical derivation guarantees small changes, this aspect is more of an algorithmic characteristic. Therefore, we propose a change tracking of eigenvalues as the eigenvalues mostly determine the decomposition.

Since the number of eigenvalues is not huge, our identification of eigenvalue pairs (old and new) is based on a similarity matrix (see Fig. 6). For the visualization, we use the representation in the complex plane and vector-like connections with dots at the end of the line that indicate the position of the eigenvalues. Due to ring-like shapes of the old eigenvalues, it is possible to identify the direction, even in the case of very small changes. To obtain a mapping between the old and new eigenvalues, pairs are identified successively on the basis of the highest value in the similarity matrix and the respective row and column are excluded after the identification. In this way, a unique mapping is guaranteed that highlights closest connections first.

For the computation of the similarity matrix  $Q$ , we use a DMD-specific formulation that uses the location of eigenvalues in the complex

plane and the modes as they are objects in the high-dimensional space with a greater variance. We assume that  $\lambda_1, \dots, \lambda_m$  and  $\vartheta_1, \dots, \vartheta_m$  are the old eigenvalues and modes, respectively, and  $\tilde{\lambda}_1, \dots, \tilde{\lambda}_m$  and  $\tilde{\vartheta}_1, \dots, \tilde{\vartheta}_m$  are the new eigenvalues and modes, respectively. While the eigenvalues can be directly compared via the absolute value, i.e.,  $\delta_{ij}^\lambda = |\lambda_i - \tilde{\lambda}_j|$ , the modes stem from eigenvectors and need an additional scaling for comparison:  $\delta_{ij}^\vartheta = \|\vartheta_i - \tilde{\vartheta}_j^* \vartheta_i (\tilde{\vartheta}_j^* \tilde{\vartheta}_j)^{-1} \cdot \tilde{\vartheta}_j\|_2$ . Normalizing the eigenvalue and eigenvector errors ( $\delta_{ij}^\lambda, \delta_{ij}^\vartheta$ ) by the maximum errors in all pairs and taking a convex combination leads to the dissimilarity matrix  $D$ . Thus, the similarity matrix  $Q$  is given by  $Q_{ij} = 1 - D_{ij}$ . The convex combination parameter can be used to emphasize the eigenvalue or mode weighting.

## 6 RESULTS

This section presents the application of constrained DMD. First, we compare our method to other spatio-temporal decompositions. Then, we apply constrained DMD to both a univariate (where external knowledge is used) and a multivariate time series (where time-scale information is used) and explain the handling from a user perspective. Another evaluation is provided in the supplemental material [34], where we investigate a time series without any external knowledge.

### 6.1 Comparison of Spatio-Temporal Decompositions

In this subsection, we compare constrained DMD with DMD, SSA, and STL. For this, we use the lynx dataset, which is a time series with numbers of annual lynx trappings in Canada from 1821–1934. The dataset consists of 114 measurements, however, to complicate the investigation, we only use the last 70 measurements, i.e., the dataset is given by  $z_0, \dots, z_{69} \in \mathbb{R}$ . The time series consists of a mixture of cyclic and seasonal patterns as well as a slightly increasing trend. For the application of SSA and DMD, we used the delay parameter  $d = 41$ , which satisfies the condition  $d \geq 69/2$ . This leads to delayed data  $x_0, \dots, x_{28} \in \mathbb{R}^{42}$  as input. STL was applied with seasons 9 and 10, and

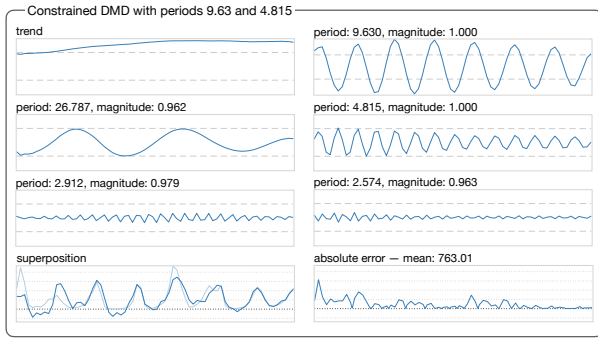


Fig. 8. Application of constrained DMD to the lynx dataset. Frequencies that belong to the periods 9.63 and 4.815, respectively, are incorporated.

constrained DMD exploits the fact that many animal populations in North Canada are periodic with a period of 9.63 [10]. Thus, constraints with periods 4.815, 9.63, 19.26, and 38.52 were incorporated (i.e., 8 user-defined frequencies are used). The results are shown in Fig. 7. The dashed lines correspond to the values 1000 and -1000 (trend and seasonal), whereas the dotted lines characterize an increase of 1500 starting with the bold dotted line at 0 (superposition and absolute error).

The first observation is that STL produces fewer components that may not identify seasonal and cyclic patterns appropriately. One reason for this is that STL cannot use non-integer numbers as input (we used 9 and 10). Another important aspect is that STL does not treat cyclic patterns separately. In fact, STL usually encodes these in the trend. In contrast, SSA, DMD, and constrained DMD produce various components. SSA sorts these by the corresponding singular values, whereas DMD and constrained DMD use our proposed filtering (starting with the trend components and followed by the seasonal components, see Fig. 5). All three methods result in a single trend (that slightly increases) and multiple oscillatory patterns. A major advantage of (constrained) DMD is the additional information of the period, which is given by the corresponding eigenvalue. This aspect enables interpretations and provides access to techniques like clustering. For constrained DMD, the structure of harmonics is clear and the clustering provides useful insights such as the periodicities in the lynx dataset (see Fig. 5). In contrast, conventional DMD mixes up relevant frequencies, not enabling a clear clustering. In the case of SSA, clustering is performed via a comparison of the corresponding singular values. However, it can be misleading, e.g., clustering the second and third component will result in a good approximation of the dominant 9.63 period, whereas clustering the fifth and sixth component results in a mixture of frequencies.

Another aspect of Fig. 7 is the reconstruction error. While a low error shows that a decomposition generally reproduces the data, it is not the primary factor for a spatio-temporal decomposition as the results may not provide insights into the data. In fact, STL has the lowest error, however, the components have less variety and mix up different periods. This demonstrates the limited applicability of STL. In contrast, SSA produces worse error yet with a better variety of patterns. Nonetheless, the interpretation of components is difficult as SSA does not naturally encode frequency into patterns (a similar incorporation approach with SSA instead of DMD would thus not be feasible). Since SSA is based on an SVD, the focus is on a best fit solution. DMD, in contrast, does a best fit with respect to a frequency-based representation. Indeed, DMD and constrained DMD produce error that are similar to SSA, but the identified patterns can be interpreted easily. While DMD fails to detect the correct frequencies, though, our constrained DMD solves this issue and produces a decomposition with a good identification and interpretation of frequencies by preserving a small error.

## 6.2 Univariate Time Series

In the following, we present a step-by-step application of our method by using the lynx dataset again. Now, we want to focus on the user perspective. Therefore, we assume to have DMD applied once, which

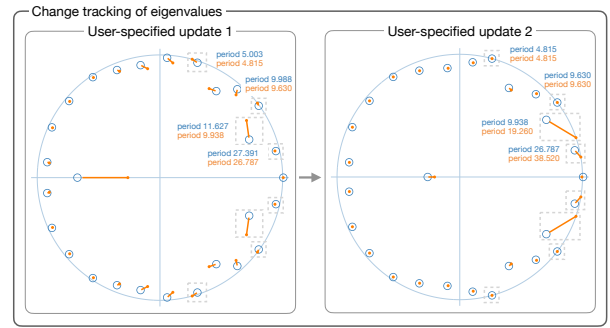


Fig. 9. Change tracking of eigenvalues of the lynx dataset.

results in 28 DMD components. According to Fig. 4, we first consult the direct visualization. Since this first impression provides not much insights (compare the similar problem in step  $\odot$  in Fig. 5), we use our filtering ( $\epsilon_T = 10^{-3}$ ,  $\epsilon_S = 0.05$ , and  $\delta = 600$ ). This leads to a clearer representation, which is illustrated in Fig. 7 (column with DMD). DMD captures the most dominant seasonal pattern via a period of 9.988. Furthermore, we see the typical behavior of harmonic patterns: besides the period 9.988, DMD identifies the period 5.003, which oscillates twice as fast. These two components could be clustered and analyzed, however, since other components seem not to feature certain patterns, we decide to give feedback to the system. Studying the literature of animal populations in North Canada leads to a period of 9.63 [10]. Hence, we use the feedback-loop via constrained DMD to incorporate the periods 9.63 and 4.815 ( $9.63/2$ ). Thus, we correct the most dominant seasonal pattern by preserving the harmonic structure.

The first user-specified update results in the same number of DMD components. Again, the direct visualization leads to a sorting of temporal evolutions, which highlights the relevant components. Consequently, we use our filtering and change tracking, which are shown in Fig. 8 and Fig. 9 (left), respectively. First of all, we successfully eliminated the period 9.988 in exchange for the period 9.63. Secondly, our approach to use the harmonic period 4.815 was useful as well since it is the third most dominant seasonal pattern now. The change tracking demonstrates that the corresponding eigenvalues are highly correlated to previous components. Indeed, the updated component with period 9.63 arises from the previous one with 9.988, and the component with period 4.815 arises from the previous one with 5.003. Since the change tracking takes the shape of temporal evolution into account (due to the similarity matrix), we observe similar oscillatory patterns.

Another observation is that our choice of user-defined periods eliminates certain mixed periods (7.358 and 4.157) except for the dominant seasonal pattern with period 26.787. The superposition (or the mean error plot) shows the effect of our update: while the error increases, our newly identified patterns lay a better foundation for the cyclic patterns. With this observation, we want to perform a second user-specified update that incorporates frequencies belonging to the cyclic behavior. As the cyclic occurs in regular intervals of four, we want to integrate the additional periods 38.52 ( $38.52/4 = 9.63$ ) and 19.26 ( $19.26/2 = 9.63$ ).

The user-specified update is shown in Fig. 5. As before, the direct visualization, i.e., sorting via influence (Fig. 5, step  $\odot$ ) barely provides insights. Our filtering (Fig. 5, step  $\star$ ) highlights more relevant components. We observe that all user-specified frequencies are included, which shows that previously dominant periods were replaced. In fact, the change tracking (see Fig. 9) reveals that the component with period 26.787 is a trade-off between the periods 19.26 and 38.52 (it is nearly the mean). According to our change tracking, the component with period 26.787 transforms to the user-defined period 38.52. A look at the superposition and error confirms that the four seasonal patterns along with the trend describe the time series adequately. Thus, we can cluster components (see Fig. 5). We observe that the trend slightly increases (except for small areas at the edge). Furthermore, superimposing certain harmonic frequencies, we can reproduce the seasonal and cyclic patterns. This observation concludes the explorative analysis.



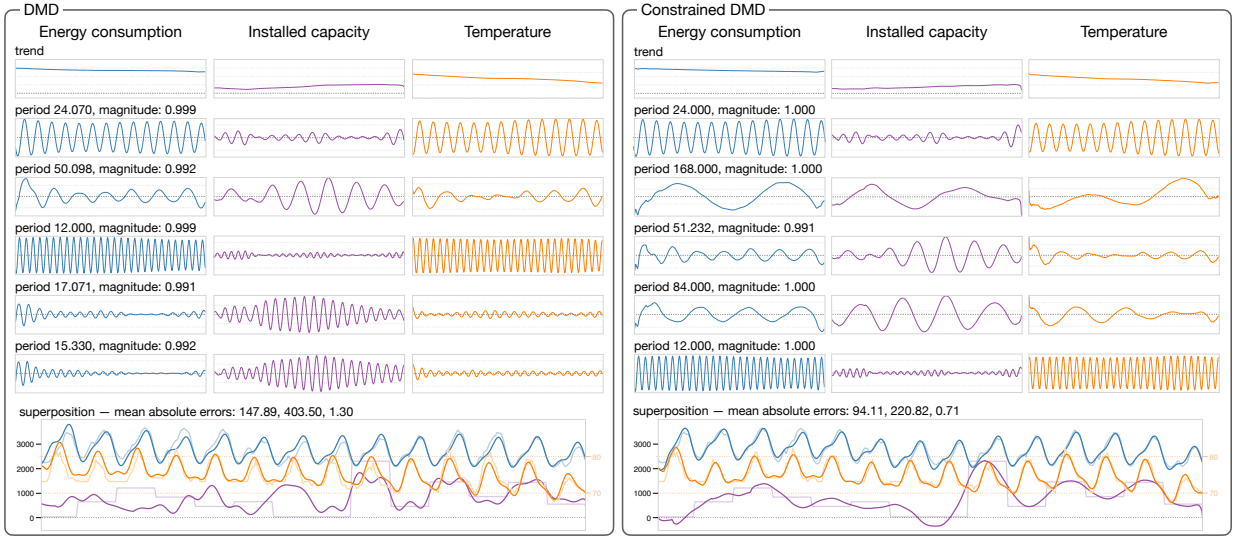


Fig. 10. Application of DMD and constrained DMD to the hourly multivariate dataset that consists of energy consumption, temperature, and the installed capacity of solar panels. Constrained DMD incorporates daily, half-daily, weekly, and half-weekly frequencies. The resulting components are filtered with regard to our proposed filtering. The scale of the energy consumption and installed capacity refers to the same values. For all three attributes, the scale of the trend matches the scale of the superposition (i.e., blue and purple start at the value 0 (bold dotted line) and increase in steps of 1000; orange start at the value 60 (bold dotted line) and increase in steps of 10). The seasonal patterns have the following scale: the bold dotted line indicates the value 0, and the other two dotted lines represent 100 and -100 (for blue and purple) and 1 and -1 (for orange), respectively.

### 6.3 Multivariate Time Series

In this subsection, a multivariate time series is analyzed from a user perspective. The time series has three attributes, where the target variable is San Diego's energy consumption (in MWh). Additionally, the highly correlated dry bulb temperature (in Fahrenheit) and installed capacity of solar panels at customer sites (in kW) are considered. The dataset is hourly and has 43,824 hours (5 years). However, we selected a section of the time series that corresponds to two weeks and, thus, our dataset is given by  $z_0, \dots, z_{336} \in \mathbb{R}^3$ . For the analysis, we used the delay parameter  $d = 203$  (it satisfies the condition  $d \geq (M - N + 1)/(N + 1) = 83.5$ ), resulting in the delayed data  $x_0, \dots, x_{133} \in \mathbb{R}^{612}$ , which is the input for (constrained) DMD. Analogously to the previous subsection, the large number of DMD components makes it hard to identify relevant patterns if only the sorting via their influence is used. Applying our filtering ( $\epsilon_T = 10^{-3}$ ,  $\epsilon_S = 0.01$ ), we obtain a representation of DMD components that is illustrated in Fig. 10 (left). The different columns show the three attributes (the scale is explained in the caption).

We see that DMD successfully identifies daily patterns with the period 24.07 and half-daily patterns with the period 12.00. According to the superposition, these two patterns (along with the trend) characterize the energy consumption and temperature quite well. For the installed capacity, the pattern with the period 50.098 is the most influential (for the energy consumption and temperature, the impact is quite small). However, the superposition shows that the shape is not accurately captured. We further observe that the shape exhibits a weekly pattern.

Thus, we can use this observation by performing a feedback-loop via constrained DMD to incorporate the period 168. Since DMD probably detect harmonics as well, we decide to integrate the period 84 as well. The result of this user-specified update is presented in Fig. 10 (right). First of all, we observe that all of our user-defined frequencies result into dominant frequencies of the system. This shows that our choice was appropriate. The resulting components of constrained DMD provide (after filtering) useful insights into the data. Namely, the update confirms our observation that the installed capacity highly exhibits a weekly pattern (and half-weekly pattern). This results in an improved superposition with a smaller mean error. Additionally, we observe that weekly patterns also enhance the superposition and mean errors of the other attributes. This concludes the use of our approach for an explorative analysis of the multivariate time series.

### 7 CONCLUSION

We have presented a mathematical formulation that enables incorporation of multiple frequencies into DMD. Based on this, an efficient algorithm and additional visualization techniques have been provided that support a human-in-the-loop process. This is a major improvement toward the interactive application with DMD. Thus, an explorative visualization and analysis is achieved, where the user is a part of the core mechanism of DMD. It is possible to bring in external knowledge or to specify identified patterns that may be wrong or only nearly correct.

In the future, we want to provide the user additional DMD-consistent tools that suggest possible frequency-based constraints. These could also partially be automatic. Another interesting research direction is the interaction of constrained DMD with the forecasting of time series. Due to our filtering and clustering, a subset of relevant DMD components is computed that may predict more accurately and transparently. In this context, it would be also interesting to apply our constrained DMD to other fields of application such as fluid dynamics or computer vision.

### ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161, and Project-ID 270852890 – GRK 2160/2.

### APPENDIX: CONSTRAINED MINIMIZATION PROBLEM

Given the matrices  $A \in \mathbb{C}^{m \times m}$  and  $\tilde{A} \in \mathbb{C}^{q \times m}$ , where  $A$  is invertible, as well as the vectors  $b \in \mathbb{C}^m$  and  $\tilde{b} \in \mathbb{C}^q$ , the solution to the constrained minimization problem  $\min_{x \in \mathbb{C}^m} \|Ax - b\|_2^2$  subject to  $\tilde{A}x = \tilde{b}$  is given by the vector  $x^* = A^{-1}b + (A^*A)^{-1}\tilde{A}^*(\tilde{A}(A^*A)^{-1}\tilde{A}^*)^{-1}(\tilde{b} - \tilde{A}A^{-1}b)$ .

*Proof.* The vector  $x^*$  trivially satisfies the constraint as  $\tilde{A}x^* = \tilde{A}A^{-1}b + \tilde{A}(A^*A)^{-1}\tilde{A}^*(\tilde{A}(A^*A)^{-1}\tilde{A}^*)^{-1}(\tilde{b} - \tilde{A}A^{-1}b) = \tilde{b}$ . Any other vector  $x \in \mathbb{C}^m$  with  $\tilde{A}x = \tilde{b}$  satisfies the following strict inequality:  $\|Ax - b\|_2^2 = \|Ax^* - b + A(x - x^*)\|_2^2 = \|Ax^* - b\|_2^2 + 2(Ax^* - b)^*A(x - x^*) + \|A(x - x^*)\|_2^2 > \|Ax^* - b\|_2^2$  as  $\|A(x - x^*)\|_2^2 > 0$  for  $x \neq x^*$  (due to the invertibility of  $A$ ) and  $(Ax^* - b)^*A(x - x^*) = (\tilde{b} - \tilde{A}A^{-1}b)^*(\tilde{A}(A^*A)^{-1}\tilde{A}^*)^{-1}\tilde{A}(A^*A)^{-1}A^*(x - x^*) = (\tilde{b} - \tilde{A}A^{-1}b)^*(\tilde{A}(A^*A)^{-1}\tilde{A}^*)^{-1}(\tilde{b} - \tilde{b}) = 0$ . This proves that the solution to the minimization problem is given by the vector  $x^*$ .  $\square$

## REFERENCES

- [1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.
- [2] M. R. Allen and L. A. Smith. Optimal filtering in singular spectrum analysis. *Physics Letters A*, 234(6):419–428, 1997. doi: 10.1016/S0375-9601(97)00559-8
- [3] B. Alsallakh, M. Bögl, T. Gschwandtner, S. Miksch, B. Esmael, A. Arnaut, G. Thonhauser, and P. Zöllner. A visual analytics approach to segmenting and labeling multivariate time series data. In *EuroVA@ Euro-Vis*, 2014. doi: 10.2312/eurova.20141142
- [4] A. M. Álvarez-Meza, C. D. Acosta-Medina, and G. Castellanos-Domínguez. Automatic singular spectrum analysis for time-series decomposition. In *European Symposium on Artificial Neural Networks*, pp. 131–136, 2013.
- [5] S. K. Badam, J. Zhao, S. Sen, N. Elmqvist, and D. Ebert. Timefork: Interactive prediction of time series. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 5409–5420. Association for Computing Machinery, 2016. doi: 10.1145/2858036.2858150
- [6] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, 1994.
- [7] J. Bernard, T. Ruppert, O. Goroll, T. May, and J. Kohlhammer. Visual-interactive preprocessing of time series data. In *Proceedings of SIGRAD 2012*, 2012.
- [8] M. W. Berry, D. Mezher, B. Philippe, and A. Sameh. Parallel algorithms for the singular value decomposition. In *Handbook of Parallel Computing and Statistics*, pp. 133–180. Chapman and Hall/CRC, 2005. doi: 10.1201/9781420028683
- [9] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2(2):165–191, 2015. doi: 10.3934/jcd.2015002
- [10] M. Bulmer. A statistical analysis of the 10-year cycle in Canada. *The Journal of Animal Ecology*, 43(3):701–718, 1974. doi: 10.2307/3532
- [11] J. V. Carlis and J. A. Konstan. Interactive visualization of serial periodic data. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST '98)*, p. 29–38. Association for Computing Machinery, 1998. doi: 10.1145/288392.288399
- [12] D. Ceneda, T. Gschwandtner, S. Miksch, and C. Tominski. Guided visual exploration of cyclical patterns in time-series. In *Proceedings of the IEEE Symposium on Visualization in Data Science (VDS)*. IEEE Computer Society, 2018.
- [13] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 143–152. IEEE, 2012. doi: 10.1109/VAST.2012.6400557
- [14] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses. *Journal of Nonlinear Science*, 22(6):887–915, 2012. doi: 10.1007/s00332-012-9130-9
- [15] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. STL: A seasonal-trend decomposition. *Journal of Official Statistics*, 6(1):3–73, 1990.
- [16] I. Cohen, O. Azencot, P. Lifshits, and G. Gilboa. Mode decomposition for homogeneous symmetric operators. *Preprint arXiv*, 2020. doi: arXiv:2007.01534
- [17] A. O. da Silva and A. Freitas. Time series components separation based on singular spectral analysis visualization: an HJ-biplot method application. *Statistics, Optimization & Information Computing*, 8(2):346–358, 2020. doi: 10.19139/soic-2310-5070-897
- [18] D. Dylewsky, D. Barajas-Solano, T. Ma, A. M. Tartakovsky, and J. N. Kutz. Stochastically forced ensemble dynamic mode decomposition for forecasting and analysis of near-periodic systems. *Preprint arXiv*, 2020. doi: arXiv:2010.04248
- [19] K. Enes, H. Errami, M. Wolter, T. Krake, B. Eberhardt, A. Weber, and J. Zimmermann. Unsupervised and generic short-term anticipation of human body motions. *Sensors*, 20(4), 2020. doi: 10.3390/s20040976
- [20] N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton. Randomized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 18(4):1867–1891, 2019. doi: 10.1137/18M1215013
- [21] K. Fujii, N. Takeishi, B. Kibushi, M. Kouzaki, and Y. Kawahara. Data-driven spectral analysis for coordinative structures in periodic human locomotion. *Scientific Reports*, 9(1):1–14, 2019. doi: 10.1038/s41598-019-53187-1
- [22] N. Golyandina and A. Zhigljavsky. *Singular Spectrum Analysis for Time Series*. Springer, 2013. doi: 10.1007/978-3-642-34913-3
- [23] D. Goswami, E. Thackray, and D. A. Paley. Constrained Ulam dynamic mode decomposition: Approximation of the Perron-Frobenius operator for deterministic and stochastic systems. *IEEE Control Systems Letters*, 2(4):809–814, 2018. doi: 10.1109/LCSYS.2018.2849552
- [24] H. Gruendl, P. Riehmman, Y. Pausch, and B. Fröhlich. Time-series plots integrated in parallel-coordinates displays. *Computer Graphics Forum*, 35(3):321–330, 2016. doi: 10.1111/cgf.12908
- [25] T. Harris and H. Yuan. Filtering and frequency interpretations of singular spectrum analysis. *Physica D: Nonlinear Phenomena*, 239(20):1958–1967, 2010. doi: 10.1016/j.physd.2010.07.005
- [26] P. Héas and C. Herzet. State-of-the-art algorithms for low-rank dynamic mode decomposition. *Preprint arXiv*, 2021. doi: arXiv:2108.09160
- [27] P. Héas and C. Herzet. Low-rank dynamic mode decomposition: An exact and tractable solution. *Journal of Nonlinear Science*, 32(1):1–24, 2022. doi: 10.1007/s00332-021-09770-w
- [28] D. Herr, F. Beck, and T. Ertl. Visual analytics for decomposing temporal event series of production lines. In *Proceedings of the 22nd International Conference Information Visualisation (IV)*, pp. 251–259. IEEE, 2018. doi: 10.1109/iV.2018.00051
- [29] A. Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016. doi: 10.1007/s40708-016-0042-6
- [30] S. Imani, S. Alaei, and E. Keogh. Putting the human in the time series analytics loop. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19)*, p. 635–644. Association for Computing Machinery, 2019. doi: 10.1145/3308560.3317308
- [31] M. Kalantari. Forecasting covid-19 pandemic using optimal singular spectrum analysis. *Chaos, Solitons & Fractals*, 142:110547, 2021. doi: 10.1016/j.chaos.2020.110547
- [32] X. Kong, C. Li, C. Wang, Y. Zhang, and J. Zhang. Short-term electrical load forecasting based on error correction using dynamic mode decomposition. *Applied Energy*, 261:114368, 2020. doi: 10.1016/j.apenergy.2019.114368
- [33] T. Krake, A. Bruhn, B. Eberhardt, and D. Weiskopf. Efficient and robust background modeling with dynamic mode decomposition. *Journal of Mathematical Imaging and Vision*, 64(4):364–378, 2022. doi: 10.1007/s10851-022-01068-0
- [34] T. Krake, D. Klötzl, B. Eberhardt, and D. Weiskopf. Constrained Dynamic Mode Decomposition - Supplemental Material, 2022. doi: 10.18419/darus-3107
- [35] T. Krake, S. Reinhardt, M. Hlawatsch, B. Eberhardt, and D. Weiskopf. Visualization and selection of dynamic mode decomposition components for unsteady flow. *Visual Informatics*, 5(3):15–27, 2021. doi: 10.1016/j.visinf.2021.06.003
- [36] T. Krake, D. Weiskopf, and B. Eberhardt. Dynamic mode decomposition: Theory and data reconstruction. *Preprint arXiv*, 2019. doi: arXiv:1909.10466
- [37] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016. doi: 10.1137/1.9781611974508
- [38] C. Maças and P. Machado. Radial calendar of consumption. In *Proceedings of the 22nd International Conference Information Visualisation (IV)*, pp. 96–102. IEEE, 2018. doi: 10.1109/iV.2018.00027
- [39] R. Maciejewski, R. Hafen, S. Rudolph, S. G. Larew, M. A. Mitchell, W. S. Cleveland, and D. S. Ebert. Forecasting hotspots—a predictive analytics approach. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):440–453, 2011. doi: 10.1109/TVCG.2010.82
- [40] X. Mao and P. Shang. Multivariate singular spectrum analysis for traffic time series. *Physica A: Statistical Mechanics and its Applications*, 526:121063, 2019. doi: 10.1016/j.physa.2019.121063
- [41] S. L. C. Martínez and J. M. Vega. Higher order dynamic mode decomposition. *SIAM Journal of Applied Dynamical Systems*, 16:882–925, 2017.
- [42] D. Neufeld. Visualization methods for periodic time series data. In *Proceedings of the Lernen, Wissen, Daten, Analysen (LWDA '21)*, pp. 163–172, 2021.
- [43] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–

- 161, 2016. doi: 10.1137/15M1013857
- [44] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164–175, 2017. doi: 10.1016/j.neucom.2017.01.105
- [45] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. doi: 10.1017/S0022112010001217
- [46] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25:249–259, 2011. doi: 10.1007/s00162-010-0203-9
- [47] G. M. Shroff. A parallel algorithm for the eigenvalues and eigenvectors of a general complex matrix. *Numerische Mathematik*, 58(1):779–805, 1990. doi: 10.1007/BF01385654
- [48] Y. Susuki and K. Sako. Data-based voltage analysis of power systems via delay embedding and extended dynamic mode decomposition. *IFAC-PapersOnLine*, 51(28):221–226, 2018. doi: 10.1016/j.ifacol.2018.11.705
- [49] F. Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, pp. 366–381. Springer, 1981.
- [50] S. Tirunagari, S. Kouchaki, N. Poh, M. Bober, and D. Windridge. Dynamic mode decomposition for univariate time series: analysing trends and forecasting. *Preprint HAL*, 2017. doi: hal-01463744
- [51] C. Tominski, J. Abello, and H. Schumann. Axes-based visualizations with radial layouts. In *Proceedings of the 2004 ACM Symposium on Applied Computing (SAC '04)*, pp. 1242–1247. Association for Computing Machinery, 2004. doi: 10.1145/967900.968153
- [52] C. Tominski and H. Schumann. Enhanced interactive spiral display. In *Proceedings of the Annual SIGRAD Conference, Special Theme: Interactivity*, number 34, pp. 53–56, 2008.
- [53] C. Tominski and H. Schumann. *Interactive Visual Data Analysis*. CRC Press, 2020. doi: 10.1201/9781315152707
- [54] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1:391–421, 2014. doi: 10.3934/jcd.2014.1.391
- [55] J. van Wijk and E. van Selow. Cluster and calendar based visualization of time series data. In *Proceedings of the 1999 IEEE Symposium on Information Visualization (InfoVis '99)*, pp. 4–9, 1999. doi: 10.1109/INFVIS.1999.801851
- [56] J. Walker, R. Borgo, and M. W. Jones. Timenotes: a study on effective chart visualization and interaction techniques for time-series data. *IEEE transactions on Visualization and Computer Graphics*, 22(1):549–558, 2015. doi: 10.1109/TVCG.2015.2467751
- [57] M. Weber, M. Alexa, and W. Müller. Visualizing time-series on spirals. In *Proceedings of the 2001 IEEE Symposium on Information Visualization (InfoVis '01)*, vol. 1, pp. 7–14, 2001. doi: 10.1109/INFVIS.2001.963273
- [58] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011. doi: 10.1109/TVCG.2011.195