Automated partial differential equation identification

Ruixian Liu ; Michael J. Bianco; Peter Gerstoft

J Acoust Soc Am 150, 2364–2374 (2021)
https://doi.org/10.1121/10.0006444

JASA ARTICLE

Automated partial differential equation identification

Ruixian Liu,[1,a)] Michael J. Bianco,[2] and Peter Gerstoft[2,b)]
[1]Department of Electrical and Computer Engineering, University of California, San Diego, California 92161, USA
[2]Scripps Institution of Oceanography, University of California San Diego, La Jolla, California 92037, USA

ABSTRACT:
Inspired by recent developments in data-driven methods for partial differential equation (PDE) estimation, we use sparse modeling techniques to automatically estimate PDEs from data. A dictionary consisting of hypothetical PDE terms is constructed using numerical differentiation. Given data, PDE terms are selected assuming a parsimonious representation, which is enforced using a sparsity constraint. Unlike previous PDE identification schemes, we make no assumptions about which PDE terms are responsible for a given field. The approach is demonstrated on synthetic and real video data, with physical phenomena governed by wave, Burgers, and Helmholtz equations. Codes are available at https://github.com/NoiseLab-RLiu/Automate-PDE-identification.
https://doi.org/10.1121/10.0006444
(Received 4 May 2021; revised 8 August 2021; accepted 8 September 2021; published online 4 October 2021)
[Editor: James F. Lynch] Pages: 2364–2374

I. INTRODUCTION

Partial differential equations (PDEs) govern many natural dynamical phenomena. Traditional methods for modeling dynamical systems with PDEs are typically based on physical principles, and analytically determining the correct PDE terms can be difficult.[1] Thus, the more applicable data-driven PDE identification methods are the subject of intensive research.

There has been significant development in data-driven PDE extraction theory thanks to the advancements in physics-informed machine learning.[1–10] Our exploration is inspired by recent work in sparse modeling.[1,2] Sparse modeling[11,12] assumes a parsimonious data representation[13,14] that scales well to big data problems and has obtained compelling results in many related fields.[15,16] Early applications of sparse, data-driven PDE estimation to real data have appeared.[17–19]

Often, we have *a priori* assumptions for the PDE and then retrieve relevant terms. In previous PDE-discovery developments, one or more active PDE terms (e.g., the first order time derivative term[1,2] or multiple PDE terms[4]) are assumed *a priori* for the PDE. The other contributing terms together with their coefficients are then derived from this prior information. Thus, only parts of the PDE are found by data-driven approaches. This can be problematic when the assumed existing term is not obvious, e.g., to identify the governing PDE for a surface wave that may either be an inviscid Burgers equation or a non-attenuating wave equation that share no PDE terms in common, one must specify the correct existing term according to sufficient prior knowledge.

To alleviate the data-driven PDE identification method's reliance on the prior information, the proposed approach can *automatically* identify all contributing terms constituting the PDE for the dynamics shown by the data. The method computes a dictionary of hypothetical PDE terms from data using finite difference (FD) and pseudo-spectral (PS) methods and selects the contributing terms using sparsity and resampling. We show that the wave, Burgers, and Helmholtz equations are well-identified from data.

II. THEORY

From a given observed field, the inverse problem solves the background parameters generating the field. Often, the inverse problem is solved under strong assumptions as only source locations are unknown, or it is a wave guide problem. The PDE generating the field has been assumed known. We relax this assumption and solve for the PDE that could have generated the observations.

A. Background

Consider a field $U(x, y, t)$ across spatial x, y and temporal t coordinates. Let it be governed by a PDE $\mathrm{N}[U(x, y, t)]$ with $f(x, y, t)$ the source term,

$$\mathrm{N}[U(x, y, t)] = f(x, y, t), \tag{1}$$

with corresponding spatial and temporal boundary conditions. We are here concerned with discovering the homogeneous PDE $\mathrm{N}[U(x, y, t)] = 0$, thus $f(x, y, t) = 0$.

Examples of $\mathrm{N}[U]$ with the typical 2–3 terms include

$$\mathrm{N}[U] = \frac{\partial^2 U}{\partial t^2} + \alpha \frac{\partial U}{\partial t} - c^2 \nabla^2 U \quad (\text{wave equation}), \tag{2}$$

$$\mathrm{N}[U] = \omega^2 U + c^2 \nabla^2 U \quad (\text{Helmholtz equation}), \tag{3}$$

a)Electronic mail: rul188@ucsd.edu, ORCID: 0000-0001-6344-5419.
b)ORCID: 0000-0002-0471-062X.

 0001-4966/2021/150(4)/2364/11/$30.00

https://doi.org/10.1121/10.0006444

$$\mathrm{N}[U] = \frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} - \nu\frac{\partial^2 U}{\partial x^2} \quad \text{(Burgers equation in one dimension)}. \tag{4}$$

In many physical environments, the exact form of $\mathrm{N}(U)$ is unknown. Consider the general form with D terms,

$$\mathrm{N}[U] = a_1\frac{\partial U}{\partial x} + a_2\frac{\partial U}{\partial y} + a_3\frac{\partial U}{\partial t} + a_4\frac{\partial^2 U}{\partial t^2} + \cdots. \tag{5}$$

Often, up to second order is assumed, but fourth order is not uncommon. Non-linear terms like $U(\partial U/\partial x)$ can appear (4), and the time derivative might be absent (3).

Consider the data of the form $\mathbf{U} \in \mathbb{C}^{N_x \times N_y \times N_t}$ for N_x horizontal, N_y vertical, and N_t temporal points, with step size Δx, Δy, and Δt. The field is generated by a physical source or perturbed initial condition and propagates through the media. We identify PDEs governing the field from the region of interest (ROI), which is $\mathbf{U}$ excluding the near-field for potentially existing sources and the spatial-temporal boundary regions where derivatives are not defined.

B. Building a dictionary

From $\mathbf{U}(i_x, i_y, i_t)$ we obtain hypothetical PDE terms by evaluating derivatives at all points in the ROI. The derivatives are estimated using numerical methods including finite difference[20] and the PS approach.[21] At every point, the homogeneous PDE like (5) is satisfied as

$$a_1 U_x(i_x, i_y, i_t) + a_2 U_y(i_x, i_y, i_t) + \cdots = 0. \tag{6}$$

In vector notation, (6) becomes

$$\begin{aligned} &\boldsymbol{\varphi}^{\mathrm{T}}(i_x, i_y, i_t)\mathbf{a} = 0\,, \quad \mathbf{a} = [a_1 \ldots a_D]^{\mathrm{T}}, \\ &\boldsymbol{\varphi}^{\mathrm{T}}(i_x, i_y, i_t) = \left[U_x(i_x, i_y, i_t)\, U_y(i_x, i_y, i_t) \cdots\right]. \end{aligned} \tag{7}$$

For all points in the ROI, we obtain

$$\boldsymbol{\Phi}\mathbf{a} = \mathbf{0}, \quad \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\varphi}^{\mathrm{T}}(1_x, 1_y, 1_t) \\ \vdots \\ \boldsymbol{\varphi}^{\mathrm{T}}((N)_x, (N)_y, (N)_t) \end{bmatrix} \in \mathbb{C}^{N \times D}, \tag{8}$$

with $N < N_x N_y N_t$, $\boldsymbol{\varphi}(i_x, i_y, i_t)$ all hypothetical PDE terms evaluated at (i_x, i_y, i_t) and $\mathbf{a} \in \mathbb{C}^D$ the PDE term coefficients. $(N)_x$ is i_x when row index $i = N$.

Each column of $\boldsymbol{\Phi}$ contains values for one PDE term evaluated at every point in the ROI. Denote the dth column as ϕ_d; from (8), we rewrite $\boldsymbol{\Phi}$ having $D = 14$ terms used for our experiments with indices shown in superscripts as

$$\begin{aligned} \boldsymbol{\Phi} &= [\,\phi_1 \cdots \phi_D] \\ &= [\overset{1}{\mathbf{1}}\ \overset{2}{\mathbf{u}_t}\ \overset{3}{\mathbf{u}_{tt}}\ \overset{4}{\mathbf{u}}\ \overset{5}{\mathbf{u}_x}\ \overset{6}{\mathbf{u}_y}\ \overset{7}{\mathbf{u}\circ\mathbf{u}_x}\ \overset{8}{\mathbf{u}\circ\mathbf{u}_y} \\ &\quad \overset{9}{\mathbf{u}_{xx}}\ \overset{10}{\mathbf{u}_{xy}}\ \overset{11}{\mathbf{u}_{yy}}\ \overset{12}{\mathbf{u}\circ\mathbf{u}_{xx}}\ \overset{13}{\mathbf{u}\circ\mathbf{u}_{xy}}\ \overset{14}{\mathbf{u}\circ\mathbf{u}_{yy}}], \end{aligned} \tag{9}$$

with $\mathbf{u} = \mathrm{vec}(\mathbf{U}) \in \mathbb{C}^N$, subscripts indicating numerical differentiation, $\mathbf{1}$ as the all-ones vector, and $\circ$ as the Hadamard (element-wise) product. $\boldsymbol{\Phi}$ contains common terms for multiple PDEs possibly governing $\mathbf{U}$ in ROI including spatial, temporal derivatives of various orders and non-linear terms. Only a few of these are in the true PDE, i.e., $\|\mathbf{a}\|_0 \ll D$ with $\|\cdot\|_0$ the number of non-zero entries.

To calculate $\boldsymbol{\Phi}$, the second order FD we use for first and second order derivatives is computed by approximating analytical derivatives using truncated Taylor series. With step Δx, its truncation error is $O(\Delta x^2)$.[20] For FD, the first and last pixels in each dimension are not considered as ROI.

The PS method[21,22] is typically more accurate than finite difference, as it is the limit of finite difference approximations when the order tends to infinity.[23] The PS is based on Fourier transform. Denote some discrete signal along the x axis for fixed y, t as $\mathbf{u}(x) = \mathbf{U}(:, y, t)$ in $\mathbb{C}^{N_x}$, with its spectral coefficients $\tilde{u}_r$ obtained by $\tilde{u}_r = (1/N_x)\sum_{j=0}^{N_x-1}\mathbf{u}(x_j)e^{-2\pi i j r/N_x}$, $i = \sqrt{-1}$; the pth order derivative is

$$\partial_x^{(p)}\mathbf{u}(x_j) = \sum_{r=-N_x/2+1}^{N_x/2} (ik_r)^p \tilde{u}_r e^{ik_r x_j}, \quad j = 0, \ldots, N_x - 1, \tag{10}$$

where the wavenumber $k_r = (2\pi/\Delta x)(r/N_x)$. To avoid issues at the spatial boundaries, Tukey windowing is used to preserve most parts of the signal. In all experiments, for each dimension, 40% of the signal is tapered and excluded from the ROI, with 20% at either end.

C. Identifying PDE terms

Beyond the assumption that the representation is parsimonious, we assume no prior intuition of which PDE terms in the library should be relevant to a given problem. The approach is data-driven as we rely on cross-validation to obtain coefficients, which is a commonly used technique in machine learning to avoid fitting noise due to redundant terms. The proposed method, which is non-recursive and free of the assumption for independently and identically distributed (i.i.d.) Gaussian noise, forms an intuitive alternative to the threshold sparse Bayesian learning approach[10] and is summarized in Algorithm 1 with details in the following.

ALGORITHM 1. PDE identification.

Input: $\boldsymbol{\Phi} = [\phi_1 \cdots \phi_D] \in \mathbb{C}^{N\times D}$, λ
Output: $\mathbf{a} = [a_1 \cdots a_D]^{\mathrm{T}} \in \mathbb{C}^D$
$\bar{\boldsymbol{\Phi}} = [\bar{\phi}_1 \cdots \bar{\phi}_D]$, where $\bar{\phi}_d = \phi_d/\|\phi_d\|_2$, $\forall d$
for $j = 1 : D$ **do**
  for $\mathrm{T} = 0 : D-1$ **do**
    $\mathrm{L}_j(\mathrm{T}) = \mathrm{CrossValid}(j, \mathrm{T})$ // Eq. (13)
  $\hat{\mathrm{T}}_j = \mathrm{argmin}_{\mathrm{T}}\mathrm{L}_j(\mathrm{T}) + \lambda \mathrm{L}_j(D-1)\mathrm{T}$
  $\hat{\mathbf{a}}_j = \mathrm{argmin}_{\bar{\mathbf{a}}_j}\|\bar{\boldsymbol{\Phi}}\bar{\mathbf{a}}_j\|_2, \mathrm{s.t.}\,\bar{a}_j = 1, \|\bar{\mathbf{a}}_j\|_0 = \hat{\mathrm{T}}_j + 1$
  $\mathrm{Err}(j) = \|\bar{\boldsymbol{\Phi}}\hat{\bar{\mathbf{a}}}_j\|_2/\|\hat{\bar{\mathbf{a}}}_j\|_2$
$\hat{j} = \mathrm{argmin}_j \mathrm{Err}(j)$ // Choose assumed term
$\hat{\mathbf{a}} = \mathrm{argmin}_{\mathbf{a}}\|\boldsymbol{\Phi}\mathbf{a}\|_2, \mathrm{s.t.}\, a_{\hat{j}} = 1, \|\mathbf{a}\|_0 = \hat{\mathrm{T}}_{\hat{j}} + 1$

Because of noise introduced by derivative computing and measurements, the equality in (8) may not hold. To enforce parsimony and avoid the trivial $\mathbf{a} = \mathbf{0}$, we assume

there is one term ϕ_j in $\boldsymbol{\Phi}$ included in the PDE and search for T other terms, thus estimating $\mathbf{a}$ by

$$\{\hat{\mathbf{a}}, \hat{j}, \hat{\mathrm{T}}\} = \arg\min_{\mathbf{a},j,\mathrm{T}} \|\boldsymbol{\Phi}\mathbf{a}\|_2, \quad \text{s.t. } a_j = 1,\ \|\mathbf{a}\|_0 = \mathrm{T}+1,\ \mathrm{T} = \arg\min_{\mathrm{T}'} \psi_j(\mathrm{T}'). \tag{11}$$

For a given j, $\|\mathbf{a}\|_0$ is chosen from sparsity-penalized cross-validation error, defined by ψ_j (14), as described next. Since D is small ($\sim 10^1$), we cycle through all D columns for j in (11) and optimize $\mathbf{a}$, T in every case. Then $\hat{\mathbf{a}}$ is selected by minimizing a normalized fitting error [defined in (15), to be discussed] over all cases.

Specifically, let $\bar{\boldsymbol{\Phi}} = [\bar{\phi}_1 \cdots \bar{\phi}_D]$ with $\bar{\phi}_d = \phi_d / \|\phi_d\|_2, \forall d$ be the normalized $\boldsymbol{\Phi}$. Under the assumption $a_j = 1$ for an arbitrary $j \in \{1, \ldots, D\}$, we solve $\bar{\mathbf{a}}_j = [\bar{a}_1, \ldots, \bar{a}_D]$ as

$$\hat{\bar{\mathbf{a}}}_j = \arg\min_{\bar{\mathbf{a}}_j} \|\bar{\boldsymbol{\Phi}}\bar{\mathbf{a}}_j\|_2^2, \quad \text{s.t. } \|\bar{\mathbf{a}}_j\|_0 = \mathrm{T}_j + 1,\ \bar{a}_j = 1. \tag{12}$$

The T_j, i.e., the number of non-zero entries other than $\bar{a}_j$ in $\bar{\mathbf{a}}_j$, is chosen using K-fold cross-validation[24] with an additional sparsity penalty. For cross-validation, we evenly divide the rows of $\bar{\boldsymbol{\Phi}}$ into K folds. The kth fold $\bar{\boldsymbol{\Phi}}^k$ is the validation data including the jth column $\bar{\phi}^k_{j\_\mathrm{val}} \in \mathbb{C}^{N/K}$ and the other columns denoted by $\bar{\boldsymbol{\Phi}}^k_{-j\_\mathrm{val}} \in \mathbb{C}^{(N/K)\times(D-1)}$. All other folds are concatenated as training data including the jth column $\bar{\phi}^k_{j\_\mathrm{tr}} \in \mathbb{C}^{[(K-1)/K]N}$ and the other columns in $\bar{\boldsymbol{\Phi}}^k_{-j\_\mathrm{tr}} \in \mathbb{C}^{[(K-1)/K]N\times(D-1)}$. For each fold k, we calculate the coefficient $\hat{\bar{\mathbf{a}}}^k_{-j\_\mathrm{tr}}(\mathrm{T})$ with T non-zero entries minimizing $\|\bar{\phi}^k_{j\_\mathrm{tr}} + \bar{\boldsymbol{\Phi}}^k_{-j\_\mathrm{tr}}\bar{\mathbf{a}}^k_{-j\_\mathrm{tr}}\|_2^2$. To solve this least squares objective using limited columns of $\bar{\boldsymbol{\Phi}}^k_{-j\_\mathrm{tr}}$, we choose the columns contributing most in the least squares solution, resulting in a threshold least squares (TLS) scheme (detailed in Appendix B). The TLS selects locations for non-zero values in $\bar{\mathbf{a}}^k_{-j\_\mathrm{tr}}$ using the entries with T largest magnitudes in the least squares solution for fitting $\bar{\phi}^k_{j\_\mathrm{tr}}$ using all columns in $\bar{\boldsymbol{\Phi}}^k_{-j\_\mathrm{tr}}$. Compared to a classic basis selection method, orthogonal matching pursuit (OMP),[25] it is non-iterative and can work better when the column in $\bar{\boldsymbol{\Phi}}^k_{-j\_\mathrm{tr}}$ most correlated to $\bar{\phi}^k_{j\_\mathrm{tr}}$ is not active, with an example given in Sec. III C. The loss for cross-validation is (here $K = 5$)

$$\mathrm{L}_j(\mathrm{T}) = \frac{1}{K}\sum_{k=1}^{K} \left\| \bar{\phi}^k_{j\_\mathrm{val}} + \bar{\boldsymbol{\Phi}}^k_{-j\_\mathrm{val}} \hat{\bar{\mathbf{a}}}^k_{-j\_\mathrm{tr}}(\mathrm{T}) \right\|_2^2. \tag{13}$$

Minimizing (13) might not give the correct sparsity due to two reasons: (i) columns in $\bar{\boldsymbol{\Phi}}$ are often coherent since they are computed from the same $\mathbf{U}$. A newly incorporated column might be well-fitted by linearly combined existing columns and thus cause $\mathrm{L}_j(\mathrm{T})$ to plateau. For example, consider non-dispersive attenuating waves $U = \mathrm{Re}(e^{-i(k(x+2y)-\omega t)} + e^{-2i(k(2x+y)-\omega t)})$ governed by (2) whose complex wavenumber $k \approx (\omega/c)[1 - (i\alpha/2\omega)]$; when $\alpha/2c \approx 0$, we have $U_t \approx -(c/3)(U_x + U_y)$ [see Eqs. (A4) and (A10) in Appendix A], causing $U_{tt} = -\alpha U_t + c^2(U_{xx} + U_{yy}) \approx -\alpha(mU_t + n(U_x + U_y)) + c^2(U_{xx} + U_{yy})$ for some non-zero m and n, i.e., $\mathrm{L}_3(3) \approx \mathrm{L}_3(5)$. (ii) After all the relevant terms are recognized, the incorporated irrelevant columns with small coefficients can fit the noise in $\bar{\phi}_j$ introduced by numerical differentiation and thus decrease $\mathrm{L}_j(\mathrm{T})$ when T already exceeds the correct sparsity.

To exclude redundant atoms, we incorporate a sparsity penalty term[26] and select the optimal sparsity as

$$\hat{\mathrm{T}}_j = \arg\min_{\mathrm{T}} \psi_j(\mathrm{T}),\ \psi_j(\mathrm{T}) = \mathrm{L}_j(\mathrm{T}) + \lambda \mathrm{L}_j(D-1)\mathrm{T}, \tag{14}$$

with $\lambda = 1$ chosen empirically working well for our data. The $\mathrm{L}_j(D-1)$ is the average fitting error (13) for all folds with all terms used.

With $\hat{\bar{\mathbf{a}}}_j$ in (12) solved by TLS using $\mathrm{T}_j = \hat{\mathrm{T}}_j$, the normalized fitting error with the range $[0, 1]$ is

$$\mathrm{Err}(j) = \|\bar{\boldsymbol{\Phi}}\hat{\bar{\mathbf{a}}}_j\|_2 / \|\hat{\bar{\mathbf{a}}}_j\|_2, \tag{15}$$

where $\mathrm{Err}(j) = 1$ indicates $\hat{\mathrm{T}}_j = 0$ and thus $\bar{\phi}_j$ cannot be fitted properly by other columns of $\bar{\boldsymbol{\Phi}}$. Repeat all the above procedures to calculate $\mathrm{Err}(j)$ for $\forall\, j = 1, \ldots, D$ and then choose $\hat{j} = \mathrm{argmin}_j \mathrm{Err}(j)$. Setting $j = \hat{j}$, $\mathrm{T} = \hat{\mathrm{T}}_{\hat{j}}$, and letting $a_j = 1$ in (11) provides $\hat{\mathbf{a}}$.

We verify this PDE identification approach by both simulation and real data experiments as will be described in Secs. III and IV.

III. SYNTHETIC EXPERIMENTS

Three sets of experiments are carried out, i.e., identifying (i) wave equations from three-dimensional (3D) spatial-temporal areas, (ii) Helmholtz equations from two-dimensional (2D) spatial areas, and (iii) Burgers equations from 2D spatial-temporal areas. Datasets used for (ii) are from the frequency components of wavefields used for (i).

A. Wave equation

The PDE identification is tested with videos $\mathbf{U}$ sampled from continuous wavefields generated by the wave equation (2). Cylindrical propagation is assumed, since we are modeling a plate. For a source f at (f_x, f_y) and field at (x_0, y_0) with a Euclidean distance d to the source, $\mathbf{U}(x_0, y_0, t) = A[e^{\mathrm{Im}(k)d}/\sqrt{d}]f[t - (d/c)]$, where k is wavenumber, c phase speed, and A amplitude. The $\mathrm{Im}(k)$ is determined by (2) (see Appendix A).

We simulate propagation similar to the metal plate in the real data. Three videos with N_x, N_y, N_t set to 100, and $\Delta x = \Delta y = 0.001$ m and $\Delta t = 10^{-6}$ s are generated with free boundaries. The source $f(t)$ is at the center $(50.5\Delta x, 50.5\Delta y)$ and formed by summing five sinusoidals, 30, 40, 50, 60, and

https://doi.org/10.1121/10.0006444

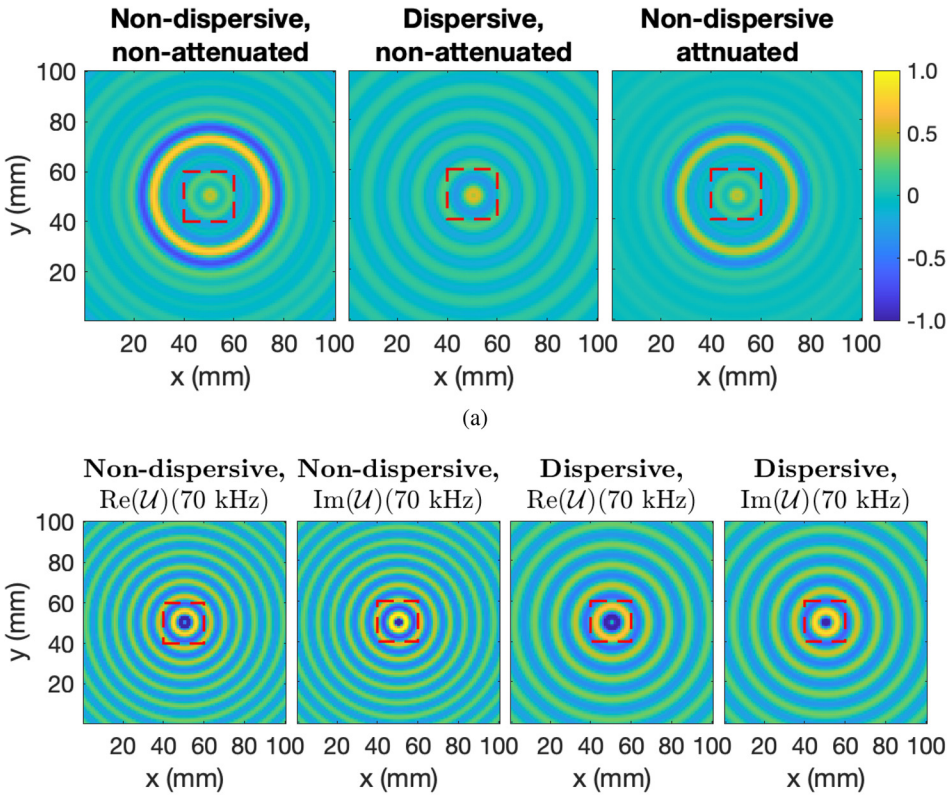

FIG. 1. (Color online) (a) Frame 50 for the synthetic wave equation. (b) Frequency components of 70 kHz for the synthetic Helmholtz equation. Pixels inside the red square are not considered.

70 kHz, with unit magnitude and zero phase at $t = 0$. Each video shows a specific wavefield, (1) non-dispersive non-attenuated wave, (2) dispersive non-attenuated wave, and (3) non-dispersive attenuated wave. The 50th frame for each of them is in Fig. 1(a).

The field $\{(i_x, i_y, i_t)|46 \le i_x \le 55; 46 \le i_y \le 55; \forall i_t\}$ near the source is dropped when extracting the PDE. We extract the PDEs for the waves at each frequency provided by a bank of ideal bandpass filters. Since $u_{tt} \approx -k^2 u$ with a constant k always holds for narrowband signals ($|\bar{\phi}_3^{\mathrm{T}} \bar{\phi}_4| > 0.99$ in our experiments), we do not consider $\mathbf{u}$ in (9).

For the non-dispersive non-attenuated waves, all the waves propagate at $c = 500\,\mathrm{m/s}$, with $\alpha = 0$. For the dispersive non-attenuated waves, the waves at 30, 40, 50, 60, and 70 kHz are with phase speeds at $c = 300$, 400, 500, 600, and 700 m/s, respectively, and $\alpha = 0$ also holds. For the non-dispersive attenuated waves, $c = 500\,\mathrm{m/s}$ and $\alpha = 2 \times 10^4$ are for all frequencies.

For all the datasets with derivatives based on both FD and PS, minimizing $\mathrm{Err}(j)$ in (15) gives $a_3 = 1$, which is for $\mathbf{u}_{tt}$. For non-attenuated waves, $\mathrm{T} = 2$ is selected by (14) with a_9 and a_{11} non-zero; for attenuated waves, $\mathrm{T} = 3$ is chosen with the non-zero entries at a_2, a_9, and a_{11}. The results are detailed in Table I, with all entries in $\mathbf{a}$ being 0 except a_2, a_3, a_9, and a_{11}. The wave equations are discovered since a_2, a_9, and a_{11} are the coefficients for $\mathbf{u}_t$, $\mathbf{u}_{xx}$, and $\mathbf{u}_{yy}$. The estimated speed $\hat{c} = \sqrt{(|a_9| + |a_{11}|)/2}$. Figure 2(a) shows $\mathrm{Err}(j)$ based on PS, and Figs. 2(b) and 2(d) where $\hat{\alpha} = a_2$ suggest the method works well as the correctly chosen PDE terms are with errors less than 5% (majority <2.5%). For a given dataset, using the PS based dictionary always provides a smaller error than using the FD based dictionary. The relation between errors and frequencies in Figs. 2(b) and 2(d) is explained as follows.

(i) The spatial-temporal differentiation works as high-pass filtering in the wavenumber-frequency domain. For PS, which computes derivatives in the wavenumber-frequency domain, an input signal of a higher frequency or wavenumber indicates a larger ratio between the derivative of the signal and the noise (from numerical differentiation), which benefits the identification. As the frequency increases, the wavenumber increases linearly for non-dispersive waves and is a constant in our dispersive waves ($100\,\mathrm{m}^{-1}$). The identified coefficients have smaller errors in both cases, and the performance improvement is larger for the non-dispersive case.

(ii) The FD computes derivatives in the spatial-temporal domain. As the period or wavelength decreases, the identification suffers from insufficient sampling. For our non-dispersive cases, both the wavelength and the period decrease for higher frequencies; the insufficient sampling is significant and leads to increasing errors. For the dispersive case, only period decreases while the wavelength is constant for larger frequencies; the benefit described in (i) is significant and results in decreasing coefficient errors.

Comparing (i) and (ii), the PS is more robust to insufficient sampling. This is due to its computing the derivatives

TABLE I. Results for synthetic wave equation recovery experiments, with "—" denoting the same values as in its upper entry. In the last four columns for each dataset, the top value in each entry is the result based on FD and the bottom is based on PS.

Frequency (kHz)	c (m/s)	α ($\times 10^4$)	a_2 ($\times 10^4$)	$-a_9$ ($\times 10^5$)	$-a_{11}$ ($\times 10^5$)	$\hat{c}$ (m/s)	c (m/s)	α ($\times 10^4$)	a_2 ($\times 10^4$)	$-a_9$ ($\times 10^5$)	$-a_{11}$ ($\times 10^5$)	$\hat{c}$ (m/s)	c (m/s)	α ($\times 10^4$)	a_2 ($\times 10^4$)	$-a_9$ ($\times 10^5$)	$-a_{11}$ ($\times 10^5$)	$\hat{c}$ (m/s)
		Non-dispersive, non-attenuated waves							Dispersive, non-attenuated waves						Non-dispersive, attenuated waves			
30	500	0	FD: 0	2.52	2.52	502	300	0	FD: 0	0.92	0.92	303	500	2	FD: 2.02	2.53	2.53	503
			PS: 0	2.52	2.52	502			PS: 0	0.90	0.90	301			PS: 2.02	2.53	2.53	503
40		—	—	2.53	2.53	503	400	—	—	1.63	1.63	404	—	—	FD: 2.03	2.53	2.53	503
				2.51	2.51	501				1.60	1.60	401			PS: 2.01	2.52	2.52	502
50		—	—	2.54	2.54	504	500	—	—	2.54	2.54	504	—	—	FD: 2.04	2.54	2.54	504
				2.51	2.51	501				2.51	2.51	501			PS: 2.01	2.51	2.51	501
60		—	—	2.55	2.55	505	600	—	—	3.64	3.64	603	—	—	FD: 2.06	2.56	2.56	505
				2.50	2.50	500				3.61	3.61	601			PS: 2.00	2.51	2.51	501
70		—	—	2.57	2.57	507	700	—	—	4.93	4.93	702	—	—	FD: 2.08	2.57	2.57	507
				2.50	2.50	500				4.91	4.91	701			PS: 2.00	2.51	2.51	501

in the wavenumber-frequency domain, implying an implicit trigonometric interpolation in the spatial-temporal domain before numerical differentiation.[27]

The proposed approach can also identify the PDE from a summation of its multiple solutions. We show this by experiments using the two unfiltered non-dispersive wavefields (attenuated and non-attenuated). Each of them is a summation of five solutions (one solution for one frequency) of one wave equation (2), with $c = 500\,\mathrm{m/s}$ and $\alpha = 0$ or 2×10^4.

Using the dictionary constructed by PS, the identified PDE for the non-attenuated waves is

$$U_{tt} - 2.50 \times 10^5 (U_{xx} + U_{yy}) = 0, \tag{16}$$

and that for the attenuated waves is

$$U_{tt} + 2.00 \times 10^4 U_t - 2.50 \times 10^5 (U_{xx} + U_{yy}) = 0. \tag{17}$$

Thus, the recovered $\hat{c} \approx 500\,\mathrm{m/s}$ for both waves, $\hat{\alpha} \approx 2 \times 10^4$ for the attenuating wave. For the dictionary based on FD, they are

$$\begin{aligned} &U_{tt} - 2.56 \times 10^5 (U_{xx} + U_{yy}) = 0, \\ &U_{tt} + 2.06 \times 10^4 U_t - 2.56 \times 10^5 (U_{xx} + U_{yy}) = 0 \end{aligned} \tag{18}$$

for the non-attenuated and attenuated waves, respectively. Thus, the recovered $\hat{c} \approx 506\,\mathrm{m/s}$ for both waves, and $\hat{\alpha} \approx 2 \times 10^4$ for the attenuating wave.

B. Helmholtz equation

Fourier transforming the $\mathbf{U}$ governed by wave equation (2) with $\alpha = 0$ at each spatial location over time, we obtain the frequency components $\mathcal{U} \in \mathbb{C}^{N_x \times N_y \times N_f}$, $N_f = N_t$. Data in each spatial frame of $\mathcal{U}$ satisfy Helmholtz equation (3). We thus use frequency components of the previous non-attenuated waves as datasets for Helmholtz equation identification.

The first spatial frame of $\mathcal{U}$ is for DC, and $\Delta f = (1/\Delta t)/N_f = 10\,\mathrm{kHz}$ between neighboring frames. Thus, we have 10 datasets used for Helmholtz equation identification, with each dataset being one frame among the fourth to eighth frames in two spectra $\mathcal{U}$, which are for non-attenuated (i) non-dispersive and (ii) dispersive waves. Figure 1(b) shows the eighth frame for both $\mathcal{U}$. The ROI on each frame excludes region $\{(i_x, i_y) | 46 \le i_x \le 55; 46 \le i_y \le 55\}$ near the source. Using the same symbol $\mathbf{U}$ to denote $\mathcal{U}$ and constructing $\mathbf{\Phi}$ as (9), since each equation is identified from a 2D frame in the frequency domain, $\mathbf{u}_t$ and $\mathbf{u}_{tt}$ are not included.

Minimizing $\mathrm{Err}(j)$, $a_4 = 1$ is selected, and thus $\mathbf{u}$ is chosen. From (14), $\mathrm{T} = 2$ is chosen for all experiments with a_9 and a_{11} non-zero, the coefficients for $\mathbf{u}_{xx}$ and $\mathbf{u}_{yy}$. The results are detailed in Table II, with all entries in $\mathbf{a}$ being 0 except a_4, a_9, and a_{11}. The PDE (3) is scaled by $1/\omega^2$ with $a_4 = 1$; thus, the estimated speed $\hat{c} = \omega\sqrt{(|a_9| + |a_{11}|)/2}$ [see Fig. 2(c)]. $\mathrm{Err}(j)$ for the 70 kHz component from dispersive waves is in Fig. 2(a). The relation between errors and frequencies in Fig. 2(c) is explained as follows.

https://doi.org/10.1121/10.0006444

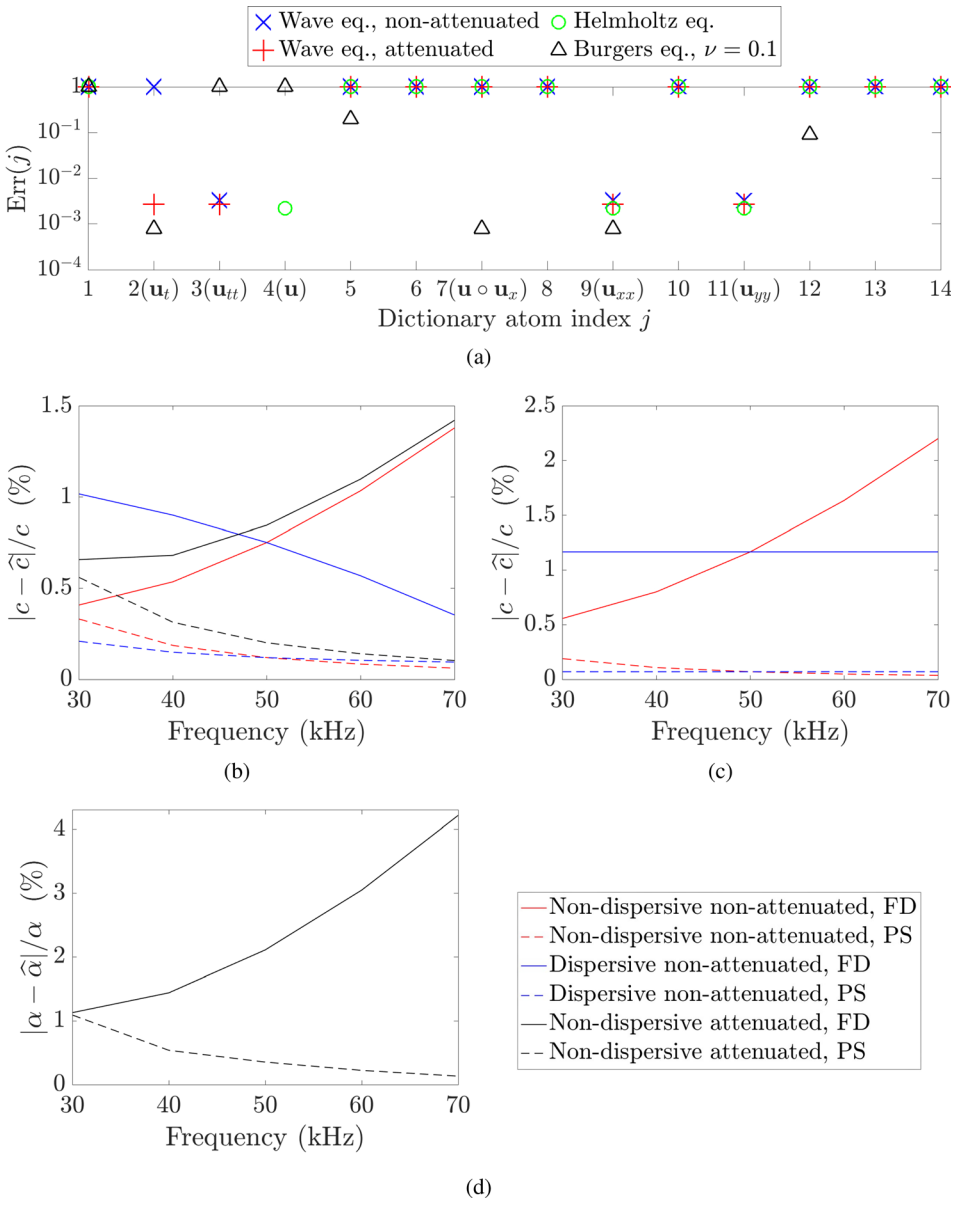

FIG. 2. (Color online) (a) Err(j) vs atom index j in $\mathbf{\Phi}$ with derivatives based on PS, including wave equations for attenuated and dispersive non-attenuated waves at 70 kHz, Helmholtz equation for the dispersive wave at 70 kHz, and Burgers equation. (b) and (c) $|c - \hat{c}|/c$ for wave equations and Helmholtz equations. (d) $|\alpha - \hat{\alpha}|/\alpha$ of attenuated waves.

(i) For Helmholtz equations, no temporal derivatives are involved, and thus the benefit for higher frequencies due to the differentiation's working as a high-pass filter disappears. But for larger wavenumbers, this benefit from differentiation still exists. This explains the decreasing errors for non-dispersive waves and the constant error for dispersive waves (whose wavenumber is a constant 100 m^{-1}) for higher frequencies using PS based dictionaries.

(ii) For the FD cases, the wavenumber (thus wavelength) is constant for all frequencies in the dispersive waves. So the same sampling sufficiency leads to a constant error. The non-dispersive waves have shorter wavelengths for higher frequencies, and the insufficient sampling issue outweighs the benefit from larger wavenumbers, causing an increasing error. Comparing it with the errors for dispersive waves (FD) in Fig. 2(b), decreasing samples spatially is more influential than decreasing samples temporally. This is because for our data the temporal sampling is more sufficient, e.g., for the 50 kHz wave propagating at 500 m/s, there are ten spatial samples in one wavelength and 20 temporal samples in one period.

C. Burgers equation

The Burgers equation (4) with viscosity ν can describe shock wave formation. We consider a one-dimensional (1D) Burgers equation on the field $\mathbf{U} \in \mathbb{R}^{500\times500}$ with $\Delta x = 0.04$ m and $\Delta t = 0.01$ s. Three fields with a same initial condition (a perturbation shaped as a Gaussian distribution PDF) governed by (4) with $\nu = 0.025$, 0.05, and 0.1 are generated by fourth order Runge–Kutta method.[28] Figure 3 shows the common initial state and the waveforms at $t = 5$ s for various ν. For a larger ν, the shock at $t = 5$ s becomes

TABLE II. Results for synthetic Helmholtz equation recovery experiments. For entries in the last three columns of each dataset, the top value is the result based on FD and the bottom is based on PS. $\omega = 2\pi \times$ Freq.

Frequency (kHz)	True c (m/s)	$\omega^2 a_9$ ($\times 10^5$)	$\omega^2 a_{11}$ ($\times 10^5$)	$\hat{c}$ (m/s)	True c (m/s)	$\omega^2 a_9$ ($\times 10^5$)	$\omega^2 a_{11}$ ($\times 10^5$)	$\hat{c}$ (m/s)
	Non-dispersive wave case				Dispersive wave case			
30	500	FD: 2.53	2.53	503	300	FD: 0.92	0.92	304
		PS: 2.51	2.51	501		PS: 0.90	0.90	300
40	—	FD: 2.54	2.54	504	400	FD: 1.64	1.64	405
		PS: 2.51	2.51	501		PS: 1.60	1.60	400
50	—	FD: 2.56	2.56	506	500	FD: 2.56	2.56	506
		PS: 2.50	2.50	500		PS: 2.50	2.50	500
60	—	FD: 2.58	2.58	508	600	FD: 3.68	3.68	607
		PS: 2.50	2.50	500		PS: 3.60	3.60	600
70	—	FD: 2.61	2.61	511	700	FD: 5.02	5.02	708
		PS: 2.50	2.50	500		PS: 4.91	4.91	700

smoother due to the increased diffusion. No source is included.

In the 1D case, terms in (9) involving derivatives along y are meaningless and thus excluded. With derivatives based on both FD and PS, $a_7 = 1$ is selected by minimizing (15) for all experiments, and thus $\mathbf{u} \circ \mathbf{u}_x$ is identified [see Fig. 2(a)]. $\mathrm{T} = 2$ is found by minimizing (14), with non-zero entries a_2, a_9 being $1.01, 1.00, 1.00$ and $-0.024, -0.05, -0.10$ for the three cases of ν based on PS. For FD, recovered a_2, a_9 are the same except that $a_2 = 1.00, a_9 = -0.025$ when $\nu = 0.025$. It works better because the spatial derivatives used for implementing the Runge–Kutta method are FD based.

If we use OMP instead of the TLS for the cross-validation and the final coefficient recovery, it will provide incorrect PDEs. Because under the assumption $\mathbf{u}_{xx}$ is active ($a_9 = 1$), the $\mathbf{u}_{tt}$ is its most correlated term and will be selected in the first iteration by OMP. The $\mathbf{u}_{tt}$ is correlated with all other terms in the dictionary, and thus incorporating $\mathbf{u}_{tt}$ into the set of the terms to fit $\mathbf{u}_{xx}$ will introduce the components of some irrelevant terms. This causes $\mathrm{L}_9(\mathrm{T})$ to become plateaued and thus $\psi_9(\mathrm{T})$ to be minimized at a sparsity larger than the correct value. With a larger $\hat{\mathrm{T}}_9$ selected, the $\mathrm{Err}(j)$ in (15) for $j = 9$ is smaller than for the other correct assumptions ($j = 2$ or 7) that have the correct $\hat{\mathrm{T}}_j$, because of more involved terms. For the Burgers equations, the $\hat{\mathrm{T}}_9 = 7, 7, 6$ using PS and 3, 6, 6 using FD for $\nu = 0.025, 0.05, 0.1$ [Fig. 4 shows the $\mathrm{L}_9(\mathrm{T})$, $\psi_9(\mathrm{T})$ for $\nu = 0.025$ with PS], and $j = 9$ always minimizes (15). After $a_9 = 1$ is assumed, since the $\hat{\mathrm{T}}_9$ is incorrect [which is supposed to be 2; see (4)], the Burgers equations cannot be identified.

Fundamentally, the OMP's only considering the most correlated atom without utilizing the relationship among all atoms in the dictionary in its first iteration leads to its failure. Unlike OMP, for $j = 9$, the TLS selects T atoms contributing most to the orthogonal projection of $\mathbf{u}_{xx}$ in the subspace spanned by all the $D - 1$ terms in $\bar{\mathbf{\Phi}}^k_{-9\_\mathrm{tr}}$. This orthogonal projection is a vector sum and is influenced by the relationship (correlation) among all vectors in the dictionary. A linear combination of $\mathbf{u}_t$ and $\mathbf{u} \circ \mathbf{u}_x$ forms the majority of the projection, and thus they are identified when $\mathrm{T} = 2$. With all the true active terms selected, the other non-zero entries in $\bar{\mathbf{a}}^k_{-9\_\mathrm{tr}}$ are found by fitting small noise in the training data, so they are of small magnitudes and work poorly in the validation data, causing $\mathrm{L}_9(\mathrm{T})$ to plateau and $\hat{\mathrm{T}}_9 = 2$ to be selected (see an example as a comparison to OMP using the same dataset in Fig. 4). So (15) is not minimized at $j = 9$ by involved irrelevant terms. In fact, $\mathrm{Err}(2) \approx \mathrm{Err}(7) \approx \mathrm{Err}(9)$, and either of the assumptions for $j = 2$, 7, or 9 leads to the correct PDE.

IV. APPLICATION TO REAL VIDEO

Our approach is demonstrated on a video of aluminum plate vibrations[29] (see Fig. 5). One period of this video considered is $\mathbf{U} \in \mathbb{R}^{100\times100\times100}$ with Δx and Δy 1 mm and sampled at 300 kHz. Vibrations contained in $\mathbf{U}$ are impulse responses for a delta function in the past; thus, no source is within the selected time.

Since aluminum plate waves are dispersive,[30] the signal is bandpass filtered to isolate wave equations for each frequency. We explore frequency bins centered from 20 to

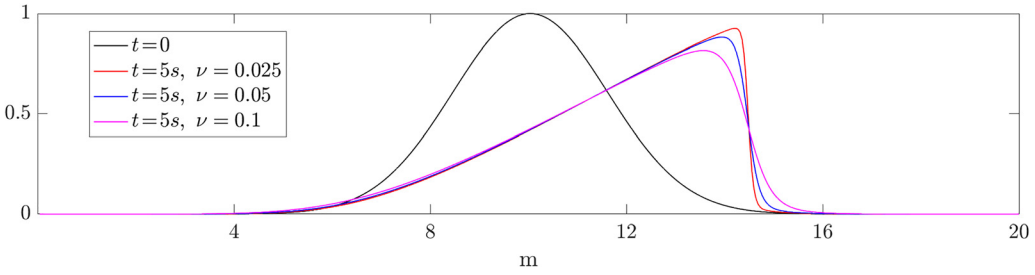

FIG. 3. (Color online) The initial state and the waveforms at $t = 5$ s corresponding to Burgers equations with various ν.

https://doi.org/10.1121/10.0006444

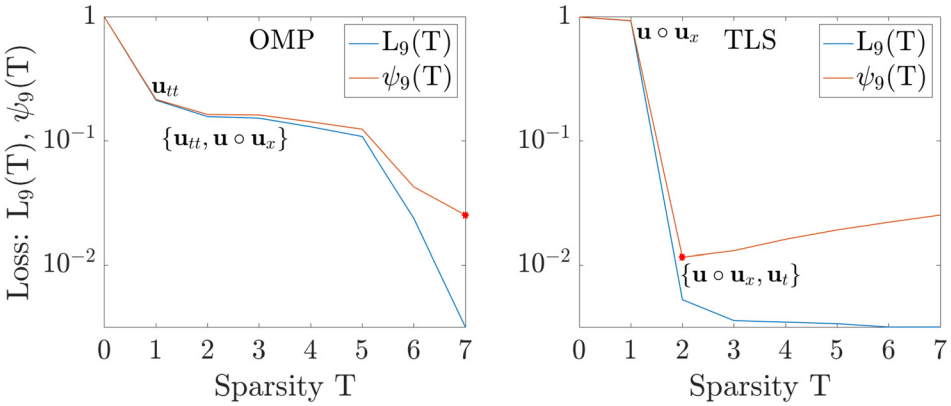

FIG. 4. (Color online) The comparison of atoms selection using TLS (right) and OMP (left) for Burgers equation identification with $\nu = 0.025$, assuming $\mathbf{u}_{xx}$ existing (i.e., $j = 9$). For a given method and a sparsity T, the selected atoms are the same for every fold in the cross-validation, and they are indicated around $\psi_9(T)$ when T is 1 or 2. The red asterisk shows the minimizer of $\psi_9(T)$.

70 kHz, with 5 kHz steps. Each bin has 1 kHz width. As in synthetic experiments, $\mathbf{u}$ is not considered for these filtered narrowband signals. $a_3 = 1$ minimizes (15) for all frequencies, with some shown in Fig. 6, and T = 2 with non-zero entries at a_9, a_{11} is always chosen by (14), as detailed in Table III. Wave equations on the plate are discovered, with $\hat{c} = \sqrt{(|a_9| + |a_{11}|)/2}$ shown in Fig. 7.

The proposed approach is compared to a classic phase speed estimation based on Fourier transform.[31,32] The method estimates phase speeds $\hat{c}_{cl}$ by finding the primary wavenumber $\hat{k}$ for each frequency f and $\hat{c}_{cl} = f/\hat{k}$. Due to the isotropic property of the wave propagation on the plate, from the wavenumber-frequency spectrum $\mathbf{K}$ of $\mathbf{U}$, for frequency f_0, $\hat{k} = \text{argmax}_k \sum_{k_x=0}^{k} |\mathbf{K}(k_x, \sqrt{k^2 - k_x^2}, f_0)|$, which finds the radius of a quarter ring with the maximal power of $\mathbf{K}$.

The underestimation by FD for high frequencies (see Fig. 7) arises from insufficient sampling along time. The PS

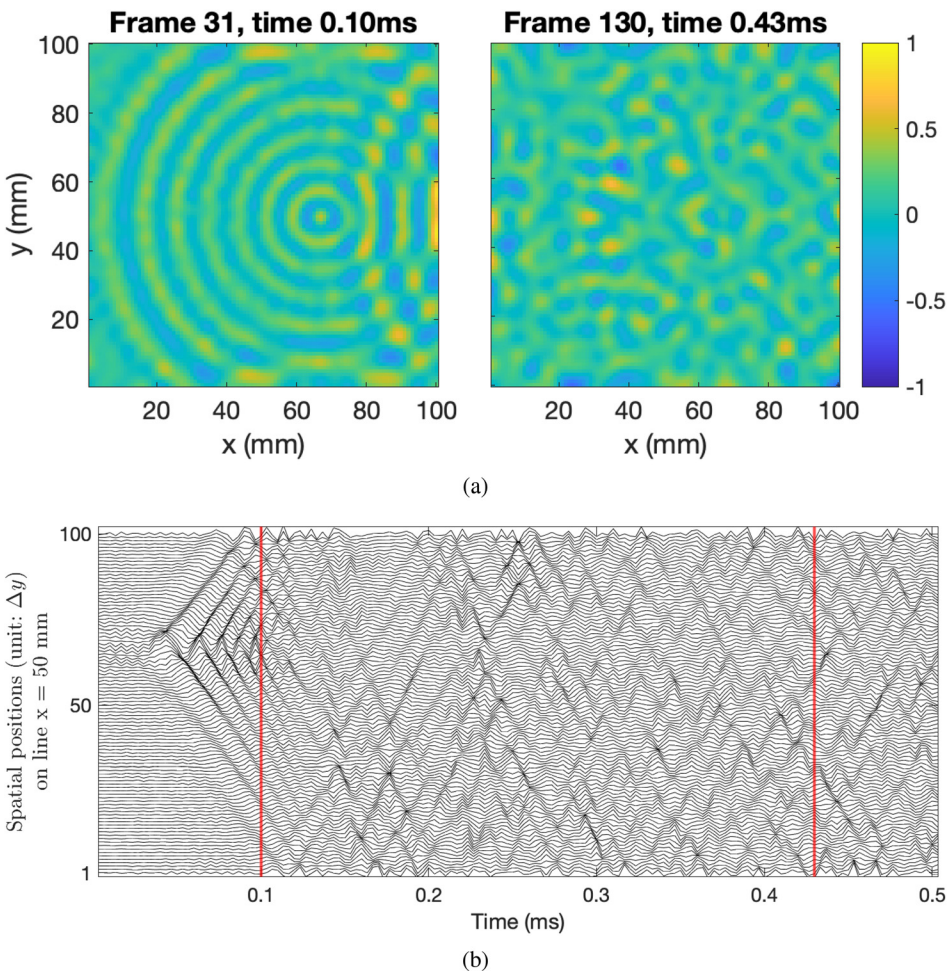

FIG. 5. (Color online) The vibrating plate: (a) the first and last selected frames, with magnitudes normalized; (b) the traces for locations at $x = 50$ mm. The selected time period is between the red lines.

TABLE III. Results for wave equation recovery on a real vibrating plate. In the columns $-a_9$, $-a_{11}$, and $\hat{c}$, the top value in each entry is the result based on FD and the bottom is based on PS.

Frequency (kHz)	$-a_9$ ($\times 10^5$)	$-a_{11}$ ($\times 10^5$)	$\hat{c}$ (m/s)	$\hat{c}_{cl}$ (m/s)
20	FD: 1.25	1.36	361	377
	PS: 1.21	1.32	355	
25	FD: 1.84	1.68	419	431
	PS: 1.92	1.65	422	
30	FD: 2.16	2.13	463	476
	PS: 2.22	2.18	469	
35	FD: 2.40	2.44	492	500
	PS: 2.40	2.49	494	
40	FD: 2.81	2.83	531	556
	PS: 2.88	3.01	543	
45	FD: 3.13	3.13	559	570
	PS: 3.32	3.19	571	
50	FD: 3.46	3.43	587	610
	PS: 3.69	3.59	603	
55	FD: 3.73	3.75	612	640
	PS: 4.04	4.06	637	
60	FD: 4.04	3.94	632	667
	PS: 4.40	4.37	662	
65	FD: 4.23	4.27	652	691
	PS: 4.74	4.81	691	
70	FD: 4.45	4.48	668	722
	PS: 5.06	5.17	715	

does trigonometric polynomial interpolation,[27] and $\partial_t^{(p)}\mathbf{u}(t_j)$ [which is calculated in the same way as $\partial_x^{(p)}\mathbf{u}(x_j)$ in (10)] is evaluated at t_j (the point) sampled from the interpolated signal. Thus, the high frequencies producing derivatives with large magnitudes are preserved. But FD evaluates $\partial_t^{(p)}\mathbf{u}(t_j)$ based on slopes of the line segments connecting $\mathbf{u}(t_j)$ with $\mathbf{u}(t_{j-1})$ and $\mathbf{u}(t_{j+1})$, respectively. When Δt is not sufficiently small, these slopes can be far from the slope of the tangent line passing $\mathbf{u}(t_j)$, causing significant bias.

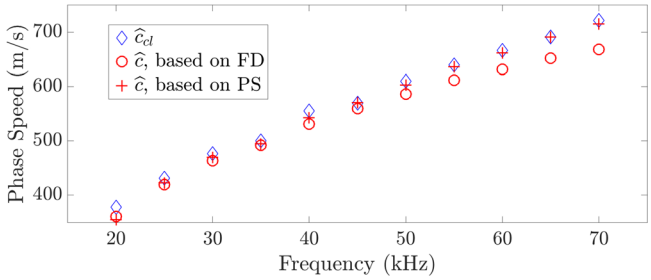

FIG. 7. (Color online) Phase speeds recovered from identified PDEs and wavenumber extraction.

V. CONCLUSION

We formulated a data-driven approach to extract PDEs without assumed terms and tested it on synthetic data and a real vibrating aluminum plate video. A dictionary containing hypothetical PDE terms is built, and correct terms are extracted by sparse modeling using cross-validation with a sparsity penalty.

APPENDIX A: WAVENUMBER DETERMINED BY WAVE EQUATIONS WITH ATTENUATION

The term $c^2\nabla^2 u$ in the wave equation indicates an isotropic propagation nature of the waves with a phase speed c. For the part of the wave that propagates along direction $\mathbf{r}$ in a circular wave, the simplified equation

$$u_{tt} = -\alpha u_t + c^2 u_{rr} + f \tag{A1}$$

can depict its dynamics without the effect of decay due to the increasing area encompassed by the wave front. For

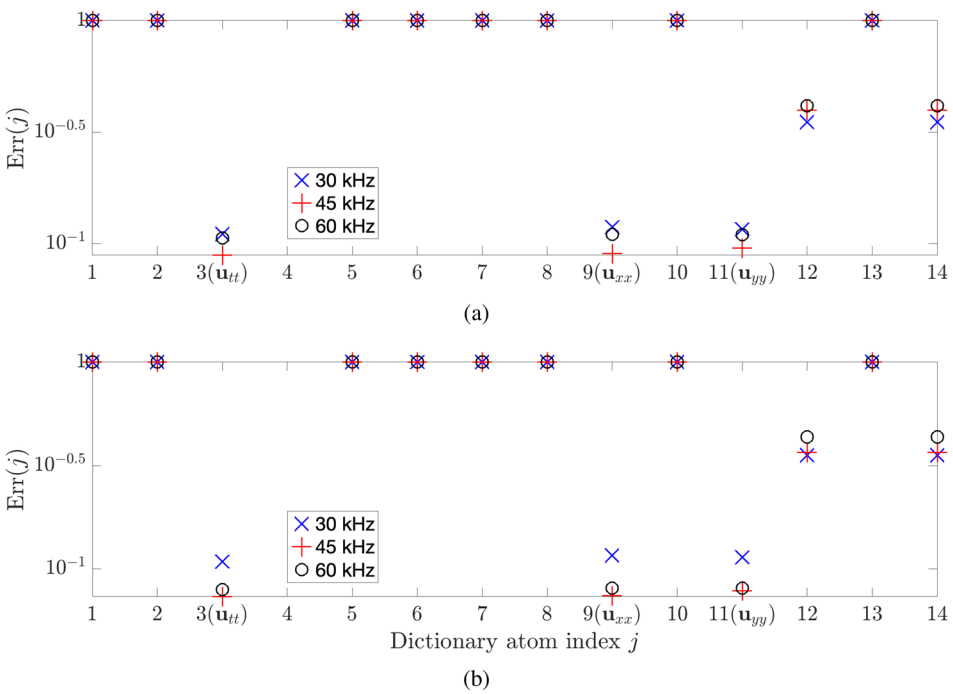

FIG. 6. (Color online) Err(j) vs atom index j in $\mathbf{\Phi}$ for three frequency bins of vibrating plate signal, with derivatives based on (a) FD and (b) PS.

https://doi.org/10.1121/10.0006444

$f=0$ and a wave at frequency ω having propagated a distance r along $\mathbf{r}$, the complex solution

$$u_c = e^{-i(kr-\omega t)} \tag{A2}$$

can satisfy (A1). Plugging (A2) into (A1) yields

$$\begin{aligned} -\omega^2 u_c + i\alpha\omega u_c + c^2 k^2 u_c &= 0, \\ -\omega^2 + i\alpha\omega + c^2 k^2 &= 0, \end{aligned} \tag{A3}$$

so

$$\begin{aligned} k^2 &= \frac{1}{c^2}(\omega^2 - i\alpha\omega) = \frac{\omega^2}{c^2}\left(1 - i\frac{\alpha}{\omega}\right), \\ k &= \frac{\omega}{c}\sqrt{1 - i\frac{\alpha}{\omega}} \approx \frac{\omega}{c}\left(1 - \frac{i\alpha}{2\omega}\right). \end{aligned} \tag{A4}$$

We can rewrite $u_c(r,t)$ as

$$u_c(r,t) = a(r,t) + ib(r,t), \tag{A5}$$

where

$$\begin{aligned} a(r,t) &= \mathrm{Re}(u_c(r,t)) \in \mathbb{R}, \\ b(r,t) &= \mathrm{Im}(u_c(r,t)) \in \mathbb{R}. \end{aligned} \tag{A6}$$

In the following equations, we abbreviate $a(r,t), b(r,t)$ as a, b, respectively.

Plug (A5) into (A1) (assume $f=0$); we have

$$\frac{\partial^2(a+ib)}{\partial t^2} + \alpha\frac{\partial(a+ib)}{\partial t} - c^2\frac{\partial^2(a+ib)}{\partial r^2} = 0, \tag{A7}$$

thus

$$\left(\frac{\partial^2 a}{\partial t^2} + \alpha\frac{\partial a}{\partial t} - c^2\frac{\partial^2 a}{\partial r^2}\right) + i\left(\frac{\partial^2 b}{\partial t^2} + \alpha\frac{\partial b}{\partial t} - c^2\frac{\partial^2 b}{\partial r^2}\right) = 0, \tag{A8}$$

and thus

$$\frac{\partial^2 a}{\partial t^2} + \alpha\frac{\partial a}{\partial t} - c^2\frac{\partial^2 a}{\partial r^2} = 0, \quad \frac{\partial^2 b}{\partial t^2} + \alpha\frac{\partial b}{\partial t} - c^2\frac{\partial^2 b}{\partial r^2} = 0, \tag{A9}$$

so $a(r,t)$ and $b(r,t)$ are both solutions for (A1).

Since the displacement field is real, we use $a(r,t)$, which is $\mathrm{Re}(u_c(r,t))$, where k is determined in (A4).

Plug (A4) into (A2); $a(r,t)$ is

$$\begin{aligned} a(r,t) &\approx \mathrm{Re}(e^{-i[(\omega/c)(1-(i\alpha/2\omega))r-\omega t]}) \\ &= e^{-(\alpha r/2c)}\cos\left(\frac{\omega}{c}r - \omega t\right). \end{aligned} \tag{A10}$$

If $\alpha r/2c$ is small across the domain, the attenuation does not contribute much to the derivatives of $a(r,t)$, causing $\partial a/\partial t \approx -c(\partial a/\partial r)$.

APPENDIX B: TLS

The TLS finds the coefficients $\mathbf{a} = [a_1 \cdots a_D]^{\mathrm{T}} \in \mathbb{C}^D$, which selects T other columns in $\mathbf{\Phi} \in \mathbb{C}^{N\times D}$ to fit its jth column. The notations here may not refer to the same variables as in the main text, for example, when in the training stage of the K-fold cross-validation, we use the $K-1$ concatenated $\bar{\mathbf{\Phi}}^k$ defined in the text as the "$\mathbf{\Phi}$" here and thus the "N" is assigned as $[(K-1)/K]N$.

First, we normalize each column of $\mathbf{\Phi}$ by its $l2$ norm and denote the normalized dictionary as $\bar{\mathbf{\Phi}} \in \mathbb{C}^{N\times D}$. For a given j, use $\bar{\mathbf{\Phi}}_{-j} \in \mathbb{C}^{N\times(D-1)}$ to denote $\bar{\mathbf{\Phi}}$ dropping its jth column $\bar{\phi}_j$, and similarly use $\mathbf{\Phi}_{-j} \in \mathbb{C}^{N\times(D-1)}$ to denote $\mathbf{\Phi}$ dropping its jth column ϕ_j. Correspondingly, we set $aj=1$ and store the other entries of $\mathbf{a}$ in $\mathbf{a}_{-j} \in \mathbb{C}^{D-1}$. The TLS is employed to compute $\mathbf{a}_{-j}$ such that $\phi_j \approx -\mathbf{\Phi}_{-j}\mathbf{a}_{-j}$ (since $\|\mathbf{\Phi a}\|_2 \approx 0$) subject to $\|\mathbf{a}_{-j}\|_0 = \mathrm{T}$.

Algorithm 2 outlines the TLS, with "†" for pseudo-inverse, "diag" for constructing a diagonal matrix from a vector, and "$\oslash$" for element-wise division (Hadamard division). Within Algorithm 2, the $\Omega = \{\Omega(1), \ldots, \Omega(\mathrm{T})\}$ denotes the set of T selected indices, e.g., if $\mathrm{T}=3$ and the three entries with maximal magnitudes in $\bar{\mathbf{a}}^{\mathrm{ls}}_{-j}$ have indices 2, 5, 7, then $\Omega = \{\Omega(1), \Omega(2), \Omega(3)\} = \{2, 5, 7\}$.

APPENDIX C: COMPARISON WITH SINDY

In the previous data-driven PDE identification method SINDy,[1] the authors used sequential threshold ridge regression (STRidge) to select active PDE terms in a normalized dictionary $\bar{\mathbf{\Phi}}_{-j} \in \mathbb{C}^{N\times(D-1)}$ (all columns have unit $l2$ norm) to fit a given PDE term ϕ_j.

The STRidge is a recursive method, where a ridge regression is implemented and the columns corresponding to small coefficients are dropped in each recursion, as illustrated in Algorithm 2. After the active terms are selected, the final coefficients are acquired by least squares regressing the assumed term ϕ_j onto the identified terms in the original dictionary (without normalization) $\mathbf{\Phi}_{-j}$.

If the correct PDE term ϕ_j is assumed, given proper λ and τ, the STRidge can work on our dataset. For example, the STRidge can recover the correct terms U_{xx} and U_{yy} for all frequencies in the Helmholtz equation dataset for dispersive waves given the correct assumption that U is an active term and $\lambda=1$, $\tau=0.1$.

ALGORITHM 2. TLS.

Input: $\mathbf{\Phi}_{-j} = [\phi_1 \cdots \phi_{j-1}\ \phi_{j+1} \cdots \phi_D] \in \mathbb{C}^{N\times(D-1)}$, ϕ_j, T
Output: $\mathbf{a}_{-j}$
$\mathbf{w}_{-j} = \{||\phi_d||_2 | \forall d, d \neq j\} \in \mathbb{C}^{D-1}$ // Column norms of $\mathbf{\Phi}$ except its jth column
$\bar{\mathbf{\Phi}}_{-j} = \mathbf{\Phi}_{-j}\mathrm{diag}^{-1}(\mathbf{w}_{-j})$ // The normalized $\mathbf{\Phi}_{-j}$ with its dth column denoted by $\bar{\phi}_d$
$\bar{\mathbf{a}}^{\mathrm{ls}}_{-j} = \bar{\mathbf{\Phi}}^{\dagger}_{-j}\phi_j$ // Least squares
$\Omega = \{\Omega(1) \ldots \Omega(\mathrm{T})\} = \{$Indices of entries in $\bar{\mathbf{a}}^{\mathrm{ls}}_{-j}$ with T maximal magnitudes$\}$
// Thresholding
$\bar{\mathbf{\Phi}}^{\mathrm{th}}_{-j} = \{\bar{\phi}_d | \forall d, d \in \Omega\} \in \mathbb{C}^{N\times \mathrm{T}}$ // Columns kept
$\bar{\mathbf{a}}^{\mathrm{th}}_{-j} = [\bar{a}^{\mathrm{th}}_1 \cdots \bar{a}^{\mathrm{th}}_{\mathrm{T}}]^{\mathrm{T}} = \bar{\mathbf{\Phi}}^{\mathrm{th}\dagger}_{-j}\phi_j$ // Least squares
$\bar{\mathbf{a}}_{-j} = [\bar{a}_1 \cdots \bar{a}_{D-1}]^{\mathrm{T}} = \mathbf{0}$ // Initialize for $\mathbf{a}_{-j}$
$\bar{a}_{\Omega(i)} = -\bar{a}^{\mathrm{th}}_i, \forall i = 1, \ldots, \mathrm{T}$ // Assign non-zero values to selected entries
$\hat{\mathbf{a}}_{-j} = \bar{\mathbf{a}}_{-j} \oslash \mathbf{w}_{-j}$ // Scaling by Hadamard division

ALGORITHM 3. Sequential threshold ridge regression (STRidge) (Ref. 1).

Input: $\bar{\boldsymbol{\Phi}}_{-j}$, ϕ_j, λ, τ, iters
Output: $\bar{\mathbf{a}}$ whose ith entry is denoted by $\bar{a}_i$
$\hat{\bar{\mathbf{a}}} = \text{argmin}_{\bar{\mathbf{a}}} ||\bar{\boldsymbol{\Phi}}_{-j}\bar{\mathbf{a}} - \phi_j||_2^2 + \lambda||\bar{\mathbf{a}}||_2^2$
bigcoeffs $= \{i : |\hat{\bar{a}}_i| \geq \tau\}$
$\hat{\bar{\mathbf{a}}}[\sim \text{bigcoeffs}] = 0$ // Threshold
$\hat{\bar{\mathbf{a}}}[\text{bigcoeffs}] = \text{STRidge}(\bar{\boldsymbol{\Phi}}_{-j}[:, \text{bigcoeffs}],$
$\phi_j, \lambda, \tau, \text{iters} - 1)$ // recursive call

If the incorrect assumed term is chosen, the SINDy cannot recognize it and will return incorrect PDE. For the same Helmholtz equation dataset, if the assumed term is U_x, then only U_{xx} is identified to be active in the dictionary.

[1]S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," Sci. Adv. **3**(4), e1602614 (2017).
[2]S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proc. Natl. Acad. Sci. U.S.A. **113**(15), 3932–3937 (2016).
[3]H. Schaeffer, G. Tran, and R. Ward, "Extracting sparse high-dimensional dynamics from limited data," SIAM J. Appl. Math. **78**(6), 3279–3295 (2018).
[4]M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).
[5]Z. Long, Y. Lu, and B. Dong, "PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network," J. Comput. Phys. **399**, 108925 (2019).
[6]S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University, Cambridge, UK, 2019).
[7]P. A. Reinbold, D. R. Gurevich, and R. O. Grigoriev, "Using noisy or incomplete data to discover models of spatiotemporal dynamics," Phys. Rev. E **101**(1), 010203 (2020).
[8]T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine, "Enforcing analytic constraints in neural networks emulating physical systems," Phys. Rev. Lett. **126**(9), 098302 (2021).
[9]W. Zhou, H. Zhang, and J. Wang, "Sparse Bayesian learning based on collaborative neurodynamic optimization," IEEE Trans. Cybern. (published online 2021).
[10]S. Zhang and G. Lin, "Robust data-driven discovery of governing physical laws with error bars," Proc. Math. Phys. Eng. Sci. **474**(2217), 20180305 (2018).
[11]M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing* (Springer Science and Business Media, New York, 2010).
[12]J. B. Harley and J. M. Moura, "Sparse recovery of the multimodal and dispersive characteristics of lamb waves," J. Acoust. Soc. Am. **133**(5), 2732–2745 (2013).
[13]J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," Proc. Natl. Acad. Sci. U.S.A. **104**(24), 9943–9948 (2007).
[14]M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," Science **324**(5923), 81–85 (2009).
[15]P. Gerstoft, C. F. Mecklenbräuker, W. Seong, and M. Bianco, "Introduction to compressive sensing in acoustics," J. Acoust. Soc. Am. **143**(6), 3731–3736 (2018).
[16]M. J. Bianco and P. Gerstoft, "Travel time tomography with adaptive dictionaries," IEEE Trans. Comput. Imag. **4**(4), 499–511 (2018).
[17]S. Khatiry Goharoodi, P. Nguyen Phuc, L. Dupré, and G. Crevecoeur, "Data-driven discovery of the heat equation in an induction machine via sparse regression," in *Proceedings of the 2019 IEEE International Conference on Industrial Technology (ICIT)*, Melbourne, Australia (February 13–15, 2019).
[18]D. Bhattacharya, L. K. Cheng, and W. Xu, "Sparse machine learning discovery of dynamic differential equation of an esophageal swallowing robot," IEEE Trans. Ind. Electron. **67**, 4711–4720 (2020).
[19]R. Liu, M. J. Bianco, and P. Gerstoft, "Wave equation extraction from a video using sparse modeling," in *Proceedings of the 53rd Asilomar Conference on Circuits, Systems and Computers*, Pacific Grove, CA (November 3–6, 2019), pp. 2160–2165.
[20]W. F. Ames, *Numerical Methods for Partial Differential Equations* (Academic, Amsterdam, 2014).
[21]J. Carcione, "A generalization of the fourier pseudospectral method," Geophysics **75**(6), A53–A56 (2010).
[22]H. Schaeffer, "Learning partial differential equations via data discovery and sparse optimization," Proc. Math. Phys. Eng. Sci. **473**(2197), 20160446 (2017).
[23]B. Fornberg, "High-order finite differences and the pseudospectral method on staggered grids," SIAM J. Numer. Anal. **27**(4), 904–918 (1990).
[24]C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006).
[25]T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," IEEE Trans. Inf. Theory **57**(7), 4680–4688 (2011).
[26]B. Qiao, X. Zhang, C. Wang, H. Zhang, and X. Chen, "Sparse regularization for force identification using dictionaries," J. Sound Vib. **368**, 71–86 (2016).
[27]B. Fornberg, "The pseudospectral method: Comparisons with finite differences for the elastic wave equation," Geophysics **52**(4), 483–501 (1987).
[28]K. Atkinson, W. Han, and D. E. Stewart, *Numerical Solution of Ordinary Differential Equations* (Wiley, New York, 2011).
[29]K. S. Alguri, J. Melville, and J. B. Harley, "Baseline-free guided wave damage detection with surrogate data and dictionary learning," J. Acoust. Soc. Am. **143**(6), 3807–3818 (2018).
[30]S. M. Ziola and M. R. Gorman, "Source location in thin plates using cross-correlation," J. Acoust. Soc. Am. **90**(5), 2551–2556 (1991).
[31]D. Alleyne and P. Cawley, "A two-dimensional Fourier transform method for the measurement of propagating multimode signals," J. Acoust. Soc. Am. **89**(3), 1159–1168 (1991).
[32]D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques* (Simon and Schuster, New York, 1992).