# AUDIO VISUALIZATION VIA DELAY EMBEDDING & SUBSPACE LEARNING

*Alois Cerbu*

Department of Mathematics
University of California, Berkeley
Berkeley, CA, US
cerbu@berkeley.edu

*Carmine Cella*

Department of Music, CNMAT
University of California, Berkeley
Berkeley, CA, US
carmine.cella@berkeley.edu

## ABSTRACT

We describe a sequence of methods for producing videos from audio signals. Our visualizations capture perceptual features like harmonicity and brightness: they produce stable images from periodic sounds and slowly-evolving images from inharmonic ones; they associate jagged shapes to brighter sounds and rounded shapes to darker ones.

We interpret our methods as adaptive FIR filterbanks and show how, for larger values of the complexity parameters, we can perform accurate frequency detection without the Fourier transform. Attached to the paper is a code repository containing the Jupyter notebook used to generate the images and videos cited. We also provide code for a realtime C++ implementation of the simplest visualization method.

We discuss the mathematical theory of our methods in the two appendices.

## 1. INTRODUCTION

In a diverse range of musical settings, it is useful to build correspondences between images and sound. Images can give instructions for creating sound – woodwind fingerings, guitar chord diagrams & tablature, traditional musical notation – or can act as proxies for its manipulation and rearrangement – such as a waveform display in a DAW's timeline, or a spectrogram in a graphic equalizer. Such methods allow sound to be experienced *offline*, out of time.

Beyond practical means for converting back and forth between sound and image, there is a rich history of audiovisual creative work – art pieces that investigate the ramifications of a fused sense of sound and sight. In this paper, we describe modes of producing images alongside sound, with the aim of augmenting or assisting auditory perception. Our methods thus blur the line between tools for analysis and creative work.

## 2. OUR METHODS

In this section we describe a sequence of methods for producing low-dimensional curves from audio signals. The methods are parametrized by choices of *embedding dimension* $N$ and *analysis dimension* $k$. All methods follow the blueprint below:

1. (*Delay embedding*) From an audio signal (a one-dimensional time series) produce an $N$-dimensional time series.
2. (*Subspace learning*) Find the $k$-dimensional subspace nearest to recent values of the high-dimensional time series.
3. (*Visualization*) Project from the $N$-dimensional space to the $k$-dimensional subspace and visualize the low-dimensional curve.

The delay embedding, popularized by Takens [1] and Packard et al. [2], is a tool for analyzing the geometry of nonlinear dynamical systems. Roughly speaking, Takens proved that the space of position-momentum pairs of a high-dimensional dynamical system (ODE) can generally be reconstructed from the trajectories of a delay-embedded one-dimensional observation of the system. The delay embedding as a visualization tool appears occasionally in the applied physics and music literature under the name *phase space reconstruction*. Gibiat [3] studied the utility of these visualizations in demonstrating certain nonlinear behaviors in feedback systems; Monro & Pressing [4], Gerhard [5] and Terez [6] used them to visualize the periodicity of musical sounds; and Lindgren et al. [7] to recognize phonemes in speech.

Our procedure for subspace learning is discussed in detail in Appendix A. In a sense, it is very similar to low-rank approximation via the truncated SVD, but, unlike a black-box numerical method, exhibits continuity properties: our estimate of subspace changes is guaranteed to change gradually with slow changes in the audio signal.

When $k = 2, 3$ the methods naturally allow us to visualize signals; when $k$ is larger, they blur into DSP methods for accurate frequency detection without the Fourier transform. Throughout, we will consider digital signals $z : \mathbf{Z} \to \mathbf{F}$, where $\mathbf{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ is the set of integers and $\mathbf{F}$ is a number system, either $\mathbf{R}$ or $\mathbf{C}$. Throughout, when $A$ is a matrix, $A^\star$ denotes its transpose or conjugate-transpose, depending on whether $A$ has real or complex entries.

### 2.1. Definitions and Preliminary Properties

Below is a more granular description of the process sketched above. First, choose the embedding dimension $N$ and projection dimension $k$. Then:

**Step 1** (Delay embedding)**.** *Given a signal $z : \mathbf{Z} \to \mathbf{F}$ and a choice of integral delays $d = (0 < d_1 < \cdots < d_{N-1})$, construct the **delay embedding** $z^d : \mathbf{Z} \to \mathbf{F}^N$, a discrete curve in $\mathbf{F}^N$ defined*

$$z^d(n) = \big(z(n), z(n - d_1), \dots, z(n - d_{N-1})\big). \tag{1}$$

**Step 2** (Subspace learning)**.** *At each timestep $n$, find an orthonormal $k$-frame $A(n)$ (an $N \times k$ matrix with orthonormal columns) whose image is the best $k$-dimensional subspace approximating*

*the discrete curve $z^d$ over recent timesteps. We find $A(n)$ by solving an optimization problem on the* ***Stiefel manifold*** *$V_k(\mathbf{F}^N)$, defined*

$$V_k(\mathbf{F}^N) = \{A \in \mathrm{Mat}_{N\times k}(\mathbf{F}) \mid A^\star A = \mathrm{Id}_k\}. \tag{2}$$

**Step 3** (Projection)**.** *Draw the curve $n \mapsto A(n)^\star z^d(n)$ in $\mathbf{F}^k$.*

In the remainder of this section, we give brief demonstrations and motivation for these visualization methods. A more thorough look at the methods, and their behavior on signals that aren't strictly periodic, is postponed to Section 3. To start, we highlight two properties of our methods:

**Observation 1.** *If $z$ is periodic then so too is $z^d$, with the same period. This means that over any time interval greater than the period, the path traveled by the curve $z^d$ will close up, and therefore the $k$-dimensional subspace best approximating $z^d$ will not change. In other words, periodic signals $z$ give stable visuals $A^\star z^d$.*

**Observation 2.** *If $z$ is nearly periodic – such as an inharmonic sum of sinusoids, or an equal-tempered chord approximating a rational interval – then $z^d$ will be as well, so $A^\star z^d$ will change slowly in shape, according to the rates that component sinusoids phase with one another.*

### 2.2. A First Pass: Minimal Embedding Dimension

In the simplest possible case, $\mathbf{F} = \mathbf{R}$, $N = k = 2$, and $d = (0, d_1)$ so the second step from 2.1 collapses. In concrete terms, we are given a real signal $z : \mathbf{Z} \to \mathbf{R}$, say with sample rate $f_s = 48\,000$. We choose $d_1$ (in this case, take $d_1 = f_s/60$) and assemble a movie with framerate $f_m = 60\,\mathrm{Hz}$ whose $j^{\text{th}}$ frame consists of a path passing through the points

$$\{(z(n), z(n - d_1))\} \subset \mathbf{R}^2. \tag{3}$$

where $n$ ranges over the time interval starting at $(f_s/f_m)j$ of length $f_s/f_m$. We make the following observations:

**Observation 3.** *When $z$ is a sinusoid of frequency $\omega$, $z^d$ will be an ellipse, whose eccentricity and orientation are determined by the relationship between $\omega$ and $f_s/d_1$.*

This first method has one principal drawback, a kind of aliasing: any periodic signal whose frequency is a multiple of $f_s/d_1$ (in our case, 60 Hz) will have visualization sitting entirely on the line $y = x$. For periodic signals with additional symmetry, like sinusoids, the aliasing appears at all harmonics of of $f_s/(2d_1)$ (in our case, 30 Hz).

In Figure 1 we show several synthetic periodic waveforms at several frequencies, visualized by this method. Notice how the shapes change as harmonics pass multiples of 60.

### 2.3. Small Embedding Dimension

We can deal with the aliasing problem by setting $N = 3$, $k = 2$ and choosing, for example, $d = (0, 491, 797)$. Since $d_1$ and $d_2 - d_1$ are coprime, no frequencies alias as in the previous example. Now our delay-embedded curve is in three-dimensional space; our subspace learning optimization algorithm figures out the best direction from which to view the space curve, and our visualization is the plane curve $n \mapsto A^\star(n)z^d(n)$.

As in the case above, sinusoids will be visualized as ellipses. More generally, for arbitrary $d = (0 < d_1 < \cdots < d_{N-1})$:

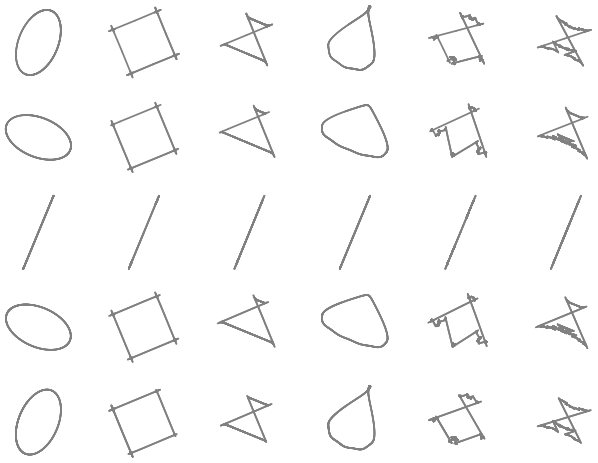

Figure 1: Visualizations of sounds of varying pitch and timbre. Each column corresponds to one waveform at various frequencies; each row to one frequency played by varying waveforms. The columns, left to right, are (a) sinusoid, (b) square, (c) sawtooth, (d) harmonic sum of sinusoids, (e) harmonic sum of square waves, (f) harmonic sum of sawtooths. The rows, top to bottom, are at fundamental frequencies 100, 110, 120, 130, 140 Hz. Notice the aliasing of all waveforms at 120 Hz.

**Observation 4.** *When $z$ is a real sinusoid $n \mapsto \sin(\omega n)$, the delay embedding $z^d$ lies on a real two-dimensional subspace of $\mathbf{R}^N$. When $z$ is a complex sinusoid $n \mapsto e^{i\omega n}$, the delay embedding $z^d$ lies on a (complex) one-dimensional subspace of $\mathbf{C}^N$ spanned by the vector*

$$\left(1, e^{-i\omega d_1}, e^{-i\omega d_2}, \ldots, e^{-i\omega d_{N-1}}\right). \tag{4}$$

The result about real sinusoids follows from the angle sum and difference identities.

In Figures 2 and 3 we visualize $A^\star z^d$ for the same waveforms as in 2.2, delay embedded in $\mathbf{R}^3$ and $\mathbf{R}^5$, respectively. To demonstrate what the subspace learning procedure is doing, we have added points to the visualization, the images of the standard basis $\{e_1, e_2, \ldots, e_N\}$ under our projection – in other words, the rows of $A$. Notice that they change position when the fundamental frequency changes. In 2.5 we will interpret $A$ as providing pitch information, while the shape of the curve encodes timbre.

### 2.4. Higher Embedding Dimension

When the embedding dimension grows, we begin to see pitch-timbre decoupling. In Figures 4 and 5, $N = 24$ and $N = 96$, respectively, we see that in a given column, the shapes of the curves $A^\star z^d$ are nearly the same, while the matrices $A$ change (as recorded in our figures by the colorful spirals).

In fact, we can interpret our algorithm as learning an FIR filterbank (the impulse responses being the columns of $A$) that captures as much of the energy of our signal as possible. When $A$ is constant, the $k$-dimensional signal $n \mapsto A^\star z^d(n)$ is precisely a convolution of $z$ with $k$ signals, each having support on $\{0, d_1, d_2, \ldots, d_{N-1}\}$.

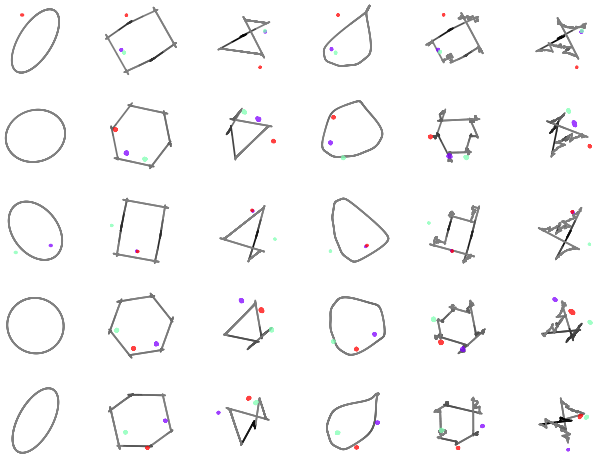

Figure 2: Visualization of the signals from Figure 1, now with $N = 3$, $d = (0, 491, 797)$. Notice the elimination of aliasing, but the variation within each column in the shapes of the gray curves. The colorful dots alongside each visualization are the rows of the projection matrix $A$.

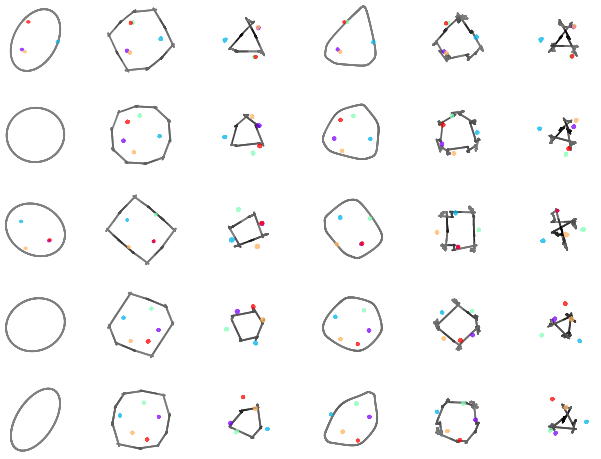

Figure 3: Visualization of the signals from Figure 1, now with $N = 5$ and $d = (0, 191, 307, 491, 797)$. Notice that shape is much more consistent in each column. The colorful dots, again, represent the rows of the learned projection matrix $A$.

Figure 4: Visualization of the signals from Figure 1, now with $N = 24$ and $d = (0, 1, 3, 6, 10, 15, \ldots, 253, 276)$. Now the shapes of the gray curves are very consistent in each column, and changes in frequency are recorded instead by the colorful spirals – the rows of $A$ – which turn through a greater angle when the pitch is higher.

### 2.5. Higher Analysis Dimension

In this section we consider methods for $k > 2$, and set $d = (0, 1, \ldots, N-1)$. The discussion is motivated by the following observation:

**Observation 5.** *The delay embedding*

$$(n \mapsto z(n)) \mapsto (n \mapsto z^d(n)), \tag{5}$$

*thought of as a map between the vector spaces of signals and $N$-dimensional curves, is linear. Therefore, following Observation 4, if $z : \mathbf{Z} \to \mathbf{F}$ is a superposition of $k$ complex (resp. real) sinusoids, the curve $z^d$ lives on a subspace of $\mathbf{F}^N$ of dimension $k$ (resp. $2k$).*

We can view Observation 5 as an improvement of Observation 1: even if our signal is not itself periodic, provided it is composed of finitely many sinusoids, the subspace learning process will stabilize, and the learned subspace will contain the information of the signal's frequency content. In the one-dimensional case, for a complex sinusoid $z : n \mapsto e^{in\omega}$, the curve $z^d$ lies on the complex line spanned by

$$v_\omega = \left(1, e^{-i\omega}, e^{-2i\omega}, \ldots, e^{-(N-1)i\omega}\right) \in \mathbf{C}^N. \tag{6}$$

If $z$ is composed of sinusoids of frequencies $\{\omega_1, \ldots, \omega_k\}$, the curve $z^d$ lives on the subspace $W$ spanned by $\{v_{\omega_1}, \ldots, v_{\omega_k}\}$. In Appendix B, we describe an algorithm that, given as input a matrix $A \in V_k(\mathbf{C}^N)$ whose image is $W$, recovers the list of frequencies. Experimentally, our algorithm is accurate to within $10^{-9}$ Hz when we set $N = 480$ and select $k = 30$ frequencies at random. The code for performing those experiments can be found in the notebook `Frequency.ipynb` in the attached repository [8].

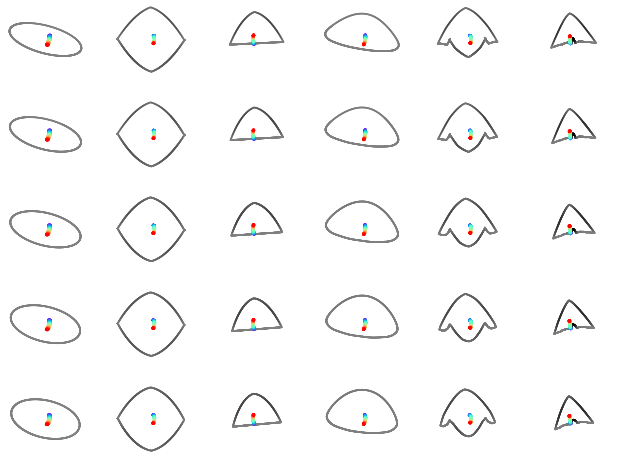

Figure 5: Visualization of the signals from Figure 1, now with $N = 96$ and $d = (0, 1, 2, 3, \ldots, 95)$. Shapes of the gray curves are almost identical in each column; again, the colorful spiral, representing rows of $A$, can be seen to tighten as the frequency increases, as in Figure 4.

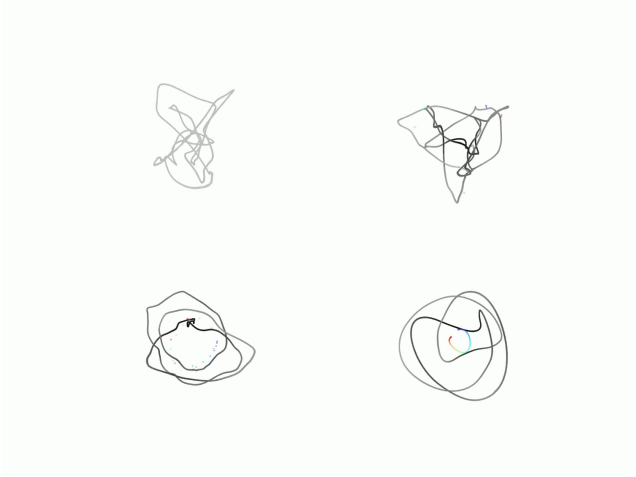

Figure 6: A still from our visualization of an accordion, playing $C3$, mezzo-forte.

Figure 7: A still from our visualizations of a double bass, playing $G2$, mezzo-forte.

## 3. VIDEO EXAMPLES & DISCUSSION

On sounds that are more complex than the simple periodic signals used to generate the static figures above, our methods produce videos as output. We provide examples of these videos in the attached code repository [8], using the parameter choices of Figures 1, 3, 4, and 5. Visualizations of orchestral sounds are found in `Orch`, and of synthetic sounds in `Synth`. In the `README` of the code repository, we have embedded a handful of videos to showcase phenomena detected by our methods.

For each sound we produce four visualizations in parallel – for those four parameter choices – to demonstrate the different information extracted in each case. For all parameter choices, periodicity of the signal is reflected in the stability of the image drawn. When the embedding dimension is small, the shape of the drawn curve is very sensitive to brightness and timbre. On the other hand, when the embedding dimension $N$ grows, the visualizations react to fundamental frequency: it's easy to distinguish by eye sounds that are high- or low-pitched: the multicolored curve representing the learned projection matrix $A$ coils more tightly for higher-pitched signals.

We generate visualizations of orchestral sounds against recordings from IRCAM's OrchideaSOL database [9]. For each of a selection of instruments (accordion, alto saxophone, bassoon, bass trombone, double bass, clarinet, flute, guitar, French horn, harp, oboe, trobone, viola, and cello) we generate videos of a sequence of notes played across the range of the instrument at three different dynamics. The purpose is to provide visual comparisons of the effects of:

- Constant pitch, varying dynamic.
- Constant pitch, varying instrument.
- Constant instrument and dynamic, varying pitch.

Our visualizations of saxophone notes see changes in the presence of high partials as amplitude increases; they see the noise & air sounds of the flute alongside its stable fundamental; visualizations of bowed double bass show that the sound is harmonic while bow is in contact with the string, and inharmonic when the bow is released (as the higher partials are no longer phase-locked by the bow's stick-slip action).

In `Synth` are visualizations of synthetic sounds:

- A harmonic sum of sinusoids with varying coefficients (constant pitch, varying brightness).
- A sum of sinusoids with varying harmonicity (constant brightness, varying harmonicity).
- White noise pushed through a resonant filter with varying $Q$ (constant pitch, varying noisiness).
- A glissando (constant timbre, varying pitch).

## 4. FURTHER WORK

### 4.1. Applications to Machine Learning

As discussed in 3, our visualizations produce images that are perceptually salient. In future work, we plan to investigate extracting features from the visualization algorithm for use in clustering, classification and other MIR-related machine learning tasks.

### 4.2. Applications to Digital Signal Processing

The frequency detection algorithm discussed in 2.5 and Appendix B performs very accurately in the synthetic cases we have tested, but there is more to study concerning its stability & robustness to noise.

Since our algorithm indirectly detects the periodicity of waveforms, our methods may have applications as well to DSP tasks like harmonic-percussive separation.

### 4.3. Applications to Performance

The methods we describe are all theoretically realtime, in the sense that they require knowledge only of present and recent values of the signal; thus they could in principle be implemented in a live audiovisual environment. We have done so already for the simplest method (from 2.2). Included in the code repository is the C++ source for that application, and a link to a performance that used it live. In future work we plan to expand this realtime application to work with larger $N$ and $k$.

Setting $\mathbf{F} = \mathbf{R}$ and $k = 3$ produces a curve representing the signal in three-dimensional space: in future work we plan to integrate our realtime waveform visualizer with a virtual reality environment or other navigable virtual three-dimensional space.

## 5. CONCLUSIONS

We have described the implementation of a sequence of methods for producing videos from sound. These methods are sensitive to various perceptual features of sound, and generalize to frequency recognition algorithms – they give visual analogs to auditory perception, and thus sit somewhere between tools for creative work and signal processing.

## 6. REFERENCES


[1] Floris Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, David Rand and Lai-Sang Young, Eds., Berlin, Heidelberg, 1981, pp. 366–381, Springer Berlin Heidelberg.

[2] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Phys. Rev. Lett.*, vol. 45, pp. 712–716, Sep 1980.

[3] V. Gibiat, "Phase space representations of acoustical musical signals," *Journal of Sound and Vibration*, vol. 123, no. 3, pp. 529–536, 1988.

[4] Gordon Monro and Jeff Pressing, "Sound visualization using embedding: The art and science of auditory autocorrelation," *Computer Music Journal*, vol. 22, no. 2, pp. 20–34, 1998.

[5] David Gerhard, "Audio visualization in phase space," in *Bridges: Mathematical Connections in Art, Music, and Science*, Reza Sarhangi, Ed., Winfield, Kansas, 1999, pp. 137–144, Southwestern College.

[6] Dmitry E. Terez, "Robust pitch determination using nonlinear state-space embedding," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002, vol. 1, pp. I–345–I–348.

[7] A.C. Lindgren, M.T. Johnson, and R.J. Povinelli, "Speech recognition using reconstructed phase space features," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, 2003, vol. 1, pp. I–60.

[8] Alois Cerbu, "Delay Embedding and Subspace Learning," `http://github.com/amcerbu/Delay-Embedding-and-Subspace-Learning`, 2024.

[9] Carmine-Emanuele Cella, Daniele Ghisi, Vincent Lostanlen, Fabien L'evy, Joshua Fineberg, and Yan Maresz, "Orchideasol: a dataset of extended instrumental techniques for computer-aided orchestration," *ArXiv*, vol. abs/2007.00763, 2020.


## A. SUBSPACE LEARNING & THE STIEFEL MANIFOLD

Let the *Stiefel manifold* $V_k(\mathbf{F}^N)$ denote the space of orthonormal $k$-frames in $\mathbf{F}^N$, that is,

$$V_k(\mathbf{F}^N) = \{A \in \mathrm{Mat}_{N\times k}(\mathbf{F}) \mid A^\star A = \mathrm{Id}_k\}. \tag{7}$$

Our subspace learning problem takes as input a segment of the delay-embedded audio signal in $\mathbf{F}^N$ (a pointcloud) and attempts to learn an orthonormal basis for a low-dimensional subspace nearby.

The space $V_k(\mathbf{F}^N)$, as its name suggests, is a compact, smooth submanifold of $\mathrm{Mat}_{N\times k}(\mathbf{F})$ whose tangent space at $A \in V_k(\mathbf{F}^N)$, which we refer to by $T_A$, has the explicit form

$$T_A = \{H \in \mathrm{Mat}_{N\times k}(\mathbf{F}) \mid A^\star H + H^\star A = 0\}. \tag{8}$$

In fact there is an explicit projection map from the ambient vector space $P_A : \mathrm{Mat}_{N\times k}(\mathbf{F}) \to T_A$ given

$$P_A : X \mapsto (\mathrm{Id} - AA^\star)X. \tag{9}$$

Geometrically, matrices tangent to $V_k(\mathbf{F}^N)$ at $A$ are those whose columns live in the orthogonal complement of $\mathrm{im}\, A$.

It is easy to check that this map is self-adjoint and idempotent with image equal to $T_A$, so it is the orthogonal projection to $T_A$. This equips us with a method of studying the problem at hand:

**Problem 1.** *Given points $y_1, \ldots, y_m \in \mathbf{F}^N$ and weights $0 \leq \alpha_j \leq 1$, find an element $A \in V_k(\mathbf{F}^N)$ minimizing*

$$f(A) = \frac{1}{2}\sum_{j=1}^{m} \alpha_j \|y_j - AA^\star y_j\|^2. \tag{10}$$

Assuming the problem can be solved tractably, a solution $A$ will provide an orthonormal basis for a $k$-dimensional approximating the data $\{y_1, \ldots, y_m\}$, weighted according to $\{\alpha_j\}$. This subspace is the best fit in the sense that it minimizes the total squared distance between points $y_j$ and their projections $AA^\star y_j$.

We solve this problem by gradient descent on $f$, and projection back to $T_A$ so as to remain near the Stiefel manifold. Expanding the expression above, we find that

$$f(A) = \frac{1}{2}\sum_{j=1}^{m} \alpha_j \big(\|y_j\|^2 - \|A^\star y_j\|^2\big), \tag{11}$$

and so, since the $\{y_j\}$ are constant, minimizing $f$ is equivalent to maximizing

$$g(A) = \frac{1}{2}\sum_{j=1}^{m} \alpha_j \|A^\star y_j\|^2. \tag{12}$$

Thinking of this as a scalar function $\mathrm{Mat}_{N\times k}(\mathbf{F}) \to \mathbf{F}$ we can compute its gradient with respect to $A$, which has the explicit form

$$\nabla g(A) = \sum_{j=1}^{m} \alpha_j \big(y_j y_j^\star\big) A = \left(\sum_{j=1}^{m} \alpha_j y_j y_j^\star\right) A. \tag{13}$$

In other words, $\nabla g(A)$ is a weighted sum of rank-one matrices.

Putting this all together, with learning rate $\epsilon > 0$, we have the following update rule:

$$A \longleftarrow A + \epsilon(\mathrm{Id} - AA^\star)\left(\sum_{j=1}^{m} \alpha_j y_j y_j^\star\right) A \tag{14}$$

This performs a step of gradient ascent on $g$, constrained to move along the tangent direction of $V_k(\mathbf{F}^N)$.

Over many iterations the matrix $A$ may drift from the Stiefel manifold. We can measure this drift by the quantity

$$h(A) = \frac{1}{4}\|A^\star A - \mathrm{Id}\|_F^2. \tag{15}$$

(The matrix norm above, $\|\cdot\|_F$, is the Frobenius norm on matrices, defined $\|B\|_F^2 = \mathrm{tr}(B^\star B)$). Clearly $h(A) = 0$ if and only if $A \in V_k(\mathbf{F}^N)$. It's not hard to compute that $\nabla h(A) = A(A^\star A - \mathrm{Id})$; with the choice of an additional learning rate $\delta > 0$, the additional update step

$$A \longleftarrow A + \delta A(\mathrm{Id} - A^\star A) = A\big((1+\delta)\,\mathrm{Id} - \delta A^\star A\big), \tag{16}$$

will keep our iteration close to the Stiefel manifold by gradient descent on $h$. We make the final observation that, since the update rule above takes the form $A \longleftarrow AX$ of a right-multiplication, we don't modify the span of the columns of $A$ via these updates: maintaining orthogonality doesn't interfere with our other iteration process, which learns the subspace.

In our code repository, this algorithm is implemented in the `analyze` method inside `Visualization.ipynb`, the main Jupyter notebook [8].

### B. FREQUENCY DETECTION

In this section we consider the case where $N$ is large. We use a full set of delays ($d = (0, 1, 2, 3, \ldots, N-1)$) and $k^2 \sim N$. For simplicity of notation we will work over $\mathbf{F} = \mathbf{C}$.

For $\omega \in [-\pi, \pi)$, recall our definition from Equation 6 of the complex vector $v_\omega = (1, e^{-i\omega}, \ldots, e^{-i(N-1)\omega})$. This vector $v_\omega$ spans the one-dimensional line of best fit for the delay embedding of the sinsuoid $z(n) = e^{i\omega n}$. We can think of $v_\omega$ as the image under $f : \mathbf{C} \to \mathbf{C}^N$

$$f(z) = \left(1, z, z^2, \ldots, z^{N-1}\right) \tag{17}$$

of the circle $S^1 \subset \mathbf{C}$ (in fancier language, $f$ is the Veronese embedding $\mathbf{CP}^1 \to \mathbf{CP}^{N-1}$ restricted to an affine chart; it generalizes the "twisted cubic").

First, I claim that for any $k \le N$, the vectors $\{v_{\omega_1}, \ldots, v_{\omega_k}\}$ are linearly independent. Construct the matrix

$$A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-i\omega_1} & e^{-i\omega_2} & \cdots & e^{-i\omega_k} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-(N-1)i\omega_1} & e^{-(N-1)i\omega_2} & \cdots & e^{-i(N-1)i\omega_k} \end{bmatrix} \tag{18}$$

This is a Vandermonde matrix; when $k \le N$ this matrix is singular iff $e^{-i\omega_j} = e^{-i\omega_\ell}$ for some $j \ne \ell$. Thus $\{v_{\omega_1}, \ldots, v_{\omega_k}\}$ is linearly independent. This has the following important consequence: if $\{\omega_1, \ldots, \omega_k\}$ and $\{\eta_1, \ldots, \eta_k\}$ are two sets of frequencies and $2k \le N$, then

$$\begin{aligned} &\mathrm{span}\{v_{\omega_1}, \ldots v_{\omega_k}\} = \mathrm{span}\{v_{\eta_1}, \ldots, v_{\eta_k}\} \\ &\implies \{\omega_1, \ldots, \omega_k\} = \{\eta_1, \ldots, \eta_k\}. \end{aligned} \tag{19}$$

In other words, subspaces of $\mathbf{C}^N$ of the form $\mathrm{span}\{v_{\omega_1}, \ldots, v_{\omega_k}\}$ determine the frequencies $\{\omega_1, \ldots, \omega_k\}$ provided $k$ isn't too large.

To retrieve frequency information from the signal $z$, we need to solve a slightly more subtle problem: our delay embedding and optimization procedures give us some orthonormal basis for $W = \mathrm{span}\{v_{\omega_j} \mid j\}$, represented by $A \in V_k(\mathbf{F}^N)$, but we want the vectors $\{v_{\omega_j}\}$ themselves.

For this discussion we augment our notation from Equation 6 to include the ambient dimension, so $v_\omega^{(N)}$ will mean the vector in $\mathbf{C}^N$ whose $n^{\text{th}}$ entry is $e^{-in\omega}$. Similarly, $W^{(N)}$ will be the span of our collection $\{v_{\omega_j}^{(N)}\}$. Consider the $(N-1) \times N$ matrices $L, R$ defined

$$L = \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix}. \tag{20}$$

Notice that $Lv_\omega^{(N)} = v_\omega^{(N-1)}$ and $Rv_\omega^{(N)} = e^{i\omega} v_\omega^{(N-1)}$, so $L, R$ both take $W^{(N)}$ surjectively to $W^{(N-1)}$.

Let $X$ be the (unknown) $N \times k$ matrix whose columns are $\{v_{\omega_j}\}$ and let $Y$ be the $k \times k$ invertible such that $AY = X$. Let $D$ be the diagonal matrix whose $j^{\text{th}}$ entry is $e^{-i\omega_j}$. Then we can summarize the behavior of $L, R$ as

$$RX = LXD^{-1}. \tag{21}$$

Now define $S = YDY^{-1}$. Then $S$ satisfies

$$\begin{aligned} RAS &= (RXY^{-1})(YDY^{-1}) \\ &= RXDY^{-1} \\ &= LXD^{-1}DY^{-1} \\ &= LXY^{-1} = LA. \end{aligned} \tag{22}$$

In other words, $S$ is the time-shift operator on $W^{(N-1)}$. It is also the solution to the least squares problem

$$(RA)^\star(RA)S = (RA)^\star LA. \tag{23}$$

which is straightforward to solve numerically since $(RA)^\star RA = A^\star(R^\star R)A$ is very close to the identity. The eigenvalues of $S$ are

the entries of $D$, and thus we've recovered the frequencies we were looking for.

The code implementing this algorithm is found in the Jupyter notebook `Frequency.ipynb` in [8].