MUSICAL ATTRACTORS: A NEW METHOD FOR AUDIO SYNTHESIS

ERIC NICHOLS[1] AND IAN KNOPKE[2]

[1] *Dept. of Computer Science, Indiana University, Bloomington, Indiana, USA*
epnichol@indiana.edu
[2] *School of Informatics, Indiana University, Bloomington, Indiana, USA*
iknopke@indiana.edu

In this paper, we use mathematical tools developed for chaos theory and time series analysis and apply them to the analysis and resynthesis of musical instruments. In particular, we can embed a basic one-dimensional audio signal time series within a higher-dimensional space to uncover the underlying generative attractor. Röbel [7],[8] described a neural-net model for audio sound synthesis based on attractor reconstruction. We present a different methodology inspired by Kaplan and Glass [3] to resynthesize the signal based on time-lag embedding in different numbers of dimensions, and suggest techniques for choosing the approximate embedding dimension to optimize the quality of the synthesized audio.

INTRODUCTION

Dynamical systems theory provides a relatively new set of techniques for analyzing systems that vary in behavior between periodic behavior and complete randomness. The field of dynamical systems grew out of a set of diverse ideas that arose in the 1960s and 70s, including such famous examples as the Lorentz attractor, Mandelbrot's Fractals, and chaos theory. Recently, these ideas have proven useful in research areas as varied as the study of fluid turbulence, weather, population behavior, and even the development of cognitive systems for controlling automation. Over time, many systems exhibit the tendency to evolve towards a particular region in state space. The regions towards which they are drawn are referred to as attractors. The overall behavior of systems that possess observable attractors can be used to identify periodic patterns in a data set that might be unobservable through more traditional statistical techniques.

In this paper we present a set of techniques for analyzing the attractors of a set of audio samples. This can in turn be used to construct a new audio signal that maintains the basic characteristics of the original sound, yet has enough variation to evade the "artificialness" that is so problematic in many synthesis systems. We consider this to be a new method of audio synthesis, with a similar function to that of frequency-modulation or physical modeling, but implemented using a newer set of techniques. The initial results are quite promising. At present, the system works best on the steady-state portion of the sounds, and we are currently working on ways to expand the range of options and audio materials that we can successfully encode.

1 BACKGROUND

While many of the techniques of dynamical systems theory require that a system to be modeled is represented in terms of a state space and transition function to move between states (such as a set of system variables, their domains, and a system of differential equations describing the evolution of the variables through time), we do not always have such explicit knowledge of the underlying dynamics. In particular, in the domain of audio signals generated with a microphone, we may only have a one-dimensional time series giving a set of sampled signal amplitudes.

Surprisingly, much information can be recovered from this one-dimensional data. If, for example, the time series has been generated by an underlying system with a higher-dimensional attractor, we can embed the time series in a higher-dimensional space (by generating vectors of time-lagged data) and reconstruct the attractor. The Takens Embedding Theorem [6] provides an upper bound on the number of dimensions needed to reconstruct the attractor. This reconstructed phase space will have a distorted shape, but can in principle be converted into the original coordinate system via a topology-preserving transformation [10]. Thus such a reconstruction will have behavior corresponding to the qualitative dynamics of the original underlying attractor.

There is good reason to expect the presence of attractors in audio data. Röbel [7] points out how a simple sound such as a sine wave with fixed period yields a one-dimensional closed loop in phase space (indeed, we made such a plot as our first test of delay-coordinate embedding). A more complex sounds such as a vibrating piano string has just one true attractor point in phase space – namely, the point representing the string at rest. However, considering just the mostly-stationary part of the signal before the amplitude decays significantly, the sound of a piano string results in a collection of periodic orbits representing the vibration frequencies. Because the harmonics of a piano string are slightly inharmonic, this idealized piano string attractor looks like a torus in phase space. We hope to find such attractors in audio signals of musical instruments.

Bernardi et al. [1] analyze several audio signals with a collection of techniques from chaos theory and dynamical systems, including attractor embedding in two or three dimensions. We perform several similar analyses of our own audio data, but the particular analyses performed are focused on determining the minimum optimal embedding dimension (which may be much higher than two or three) and properties of any chaotic attractors found in the data.

Röbel [7],[8] described a neural-net model for audio sound synthesis based on attractor reconstruction. We implement a much simpler system inspired by [3] to synthesize audio based on time-lag embedding in different numbers of dimensions, and suggest how listening to the reconstructed audio can give an empirical means to approximate the embedding dimension for musical attractors.

2 METHODOLOGY

The technique presented here resynthesizes audio from sound samples, based on the optimal embedding dimension and time lag of the time series data. The resynthesis system posits the existence of an underlying high-dimensional attractor. Hence, we developed a set of tools to determine if such an attractor exists in the given signal and to aid in optimizing the parameters for reconstructing the attractor from the original analyzed instrument. We also compare a series of support algorithms to calculate these embedding parameters. Because many of these support algorithms have their own sets of parameters that need to be tuned to work optimally, we have developed a user interface to allow experimentation and optimization of the various parameters.

The system works as follows: for each input audio sample, we perform a set of analyses to develop an understanding of the dynamics. Our first step is to choose the time lag for the delay-coordinate embedding using an autocorrelation analysis. We then calculate the largest Lyapunov exponent of the attractor and then estimate a minimal embedding dimension via the method of false nearest neighbors. We then test for the presence of an underlying deterministic system that generated the audio time series, by verifying the presence of unstable periodic orbits and testing a simple nearest-neighbor predictor. We also calculate the correlation dimension, which is equivalent to the information dimension, and look for likely unstable periodic orbits with the LK algorithm.

Finally, we construct a sound synthesizer by extending the predictor. We use several audio data points as a seed and then apply the predictor recursively to its own output, generating audio data. Note that this resynthesis algorithm is not constrained by the initial instrument sample; the output data can be a sound of arbitrary length. The sound produced shows subtle variations over time that are similar to the properties of the source instrument, in contrast to the usual sample-based looping used in most forms of commercial waveform synthesis.

2.1 Autocorrelation to estimate time lag

In delay-coordinate embedding, we require a choice of lag time between coordinates so that the separate dimensions are not too strongly related to each other (which they are if samples are taken close together in time), but also not too far apart to be completely unrelated. A common method for choosing an intermediate, useful lag time is to calculate the autocorrelation function of the series (which ranges from 1 to -1) and choose a) the smallest lag time such that this function drops to 0, b) the "autocorrelation time", where the function drops to 1/e (0.37), or else the first local minimum of the function [9]. This seems to be a bit of an art, as are many of the methods discussed here. We chose criteria a) and looked for the first value of lag near 0 (this is still a fuzzy criterion, since the meaning of "near 0" is undefined – we used visual inspection of the function to choose a reasonable value "near 0".)

2.2 Test for determinism via prediction

Next, we test for the presence of an underlying deterministic system that generated the time series, by building a simple predictor based on half of the time series data and evaluating its accuracy over the other half [1]. The predictor simply generates an embedding with the specified dimension (k) and lag (t), and then looks for the nearest neighbor elsewhere in the time series. If the given point has coordinates $\{i, i+t, i+2t, \ldots, i+(k-1)t\}$ and the nearest neighbor is $\{j, j+t, j+2t, \ldots j+(k-1)t\}$, the index $j+kt$ is used as an estimate of index $i+kt$. We tried extending this method to average the predictions of K nearest neighbors as suggested by Bernardi et al., but this was unsuccessful and resulted in signals that looked and sounded very different from the original. Predictions are generated for the test data and a normalized error metric (sum error / data series variance) is calculated.

2.3 Sound Resynthesis

We construct a sound synthesizer with this method by seeding the method with the first several (embedding dimension * lag) data points and then applying the predictor recursively to its own output. Listening to the output gives another hint as to the attractor dimension.

2.4 Embedding Dimension via False Nearest Neighbor algorithm

A method for estimating the embedding dimension for an attractor is given in [4]. The idea is to find all the neighbors (within neighborhood epsilon) of each point in the series when embedded with dimension 2. Then the series is re-embedded with dimension 3, and each point's set of neighbors is again considered. If a neighbor at dimension 2 is no longer a neighbor at dimension 3, it is considered a false nearest neighbor and added to a tally. This count is performed for successive dimensions. The percentage of false nearest neighbors at each dimension is plotted, and the graph analyzed for a hint about the proper embedding dimension (such as a local minimum or a small value of the false NN percentage.) We found that the method gave plots that simply decreased with increasing dimension; we didn't have local minima, in contrast with the algorithm examples from [4]. Also, the graphs were highly dependent on the choice of epsilon, so we found the results difficult to interpret; perhaps the method works better for chaotic attractors.

2.5 Correlation Integral to estimate dimension

The correlation dimension can be calculated quite easily, using an algorithm described in [9]. As in the false NN algorithm, here the number of nearest neighbors for each point is calculated. For a range of epsilon value, the total number of neighbors for the data set, is tabulated and expressed as a percentage of the total possible number of neighbors (i.e. the limit as epsilon becomes large and neighborhoods include all points in the set). A log-log plot is generated for this correlation sum over the range of epsilon values. Several of these plots are generated, for several values of embedding dimension. Finally, the "scaling region" of the plot (a range of epsilon values) is selected where the plots seem linear, and slopes are calculated. These slopes are values of the correlation dimension. Finally, a graph of correlation dimension vs. embedding dimension is generated. Trending towards a stable value indicates the presence of an attractor, while continually increasing correlation dimension indicates noise.

2.6 Detection of periods of unstable periodic orbits

Lathrop and Kostelich (LK) developed an algorhtm to detect unstable periodic orbits from a chaotic time series [10]. The method is simple: one generates a delay-coordinate embedding and then follows trajectories until they return to a neighborhood of the initial point. The time required is called the recurrence time, and although it might not correspond to a periodic orbit, it is likely to be such an indicator. Generating a histogram of recurrence times and looking for peaks we can identify likely orbit periods. The choice of epsilon (neighborhood radius) is again important; once the radius is too large, the histogram reaches "saturation" and loses most of its information. A radius just below this value is selected to maximize the value of the plot.

It may be the case that periods of orbits in audio correspond roughly to interesting frequency-related components, such as the fundamental pitch frequency and its overtones, as well as other frequency-related parameters such as vibrato rate. Additionally, we can theoretically look for features such as period-doubling routes to chaos as tone generation parameters are varied.

2.7 Testing for chaos (Lyapunov exponents)

2.7.1 Estimate of largest Lyapunov exponent

We implemented an algorithm from [6] to estimate the largest Lyapunov exponent of the attractor. The method is an adaptation of the standard integration method to time series data. Two nearby trajectories are compared to compute the divergence of an initial error, but when the trajectories diverge too much, a modified renormalization procedure is used to pick a new nearby trajectory before continuing.

2.7.2 Lyapunov Dimension

Algorithms exist to compute all the Lyapunov exponents from an embedded time series – however, the implementation is somewhat involved so we did not implement such an algorithm. Moreover, in our data the largest Lyapunov exponents seemed to be negative, so we did not expect to get any more information from additional exponents.

In general, however, if we did calculate all the exponents we would have another method in addition to the correlation integral and false-nearest-neighbors algorithm to estimate embedding dimension: the Lyapunov dimension is calculated by summing Lyapunov exponents, ordered by magnitude, until the sum becomes negative. The Lyapunov dimension is the maximum index of these ordered exponents such that the sum is nonnegative. The Kaplan-Yorke Conjecture [6] claims that this Lyapunov dimension is equal to the information dimension. This gives a simple way to calculate attractor dimension from the Lyapunov exponents, because the dimension can be used with the Takens embedding theorem to give an upper bound on the embedding dimension required.

It is unclear what this means for our data – the Lyapunov dimension seems to be equal to 0 because the largest exponent is negative. However, we seem to have a non-0 dimension attractor for each data set, with embedding dimension greater than 2. Either we have miscalculated Lyapunov exponents, incorrectly inferred the presence of attractors in the data, the Takens embedding theorem is not valid when the attractor dimension used is the information dimension, or we are not fulfilling the conditions of the Kaplan-Yorke Conjecture.

2.8 A note on circularity of methods

Interestingly, several of the methods one would like to use to estimate embedding dimension (such as the correlation integral and Lyapunov exponent calculation) require a choice of embedding dimension to function properly! The typical procedure is to start with an embedding dimension of 2 for these methods, and then run each analysis for successively larger embedding dimension values, up to, say, 10 dimensions, and plot the resulting curves. For example, recall the discussion of correlation dimension, which is typical for many of the methods we found: a plot of embedding dimension (x-axis) vs. correlation dimension (y-axis) will hopefully yield a curve where the correlation dimension increases and then flattens out with increasing embedding dimension; an embedding dimension point may be chosen where this plot looks asymptotic and indicates the presence of an attractor of that dimensionality [9].

3 DATA

A number of different instrument recordings were tested for this project, and in the end we chose to work with a set of four different sounds. Three clarinet tones were used: a regular clarinet note, a synthesized clarinet note, and a clarinet multiphonic. A saxophone sample was also chosen as a basis of comparison with the results of a similar analysis from [1]. All sounds analyzed were chosen from steady, sustained portions of the sounds (thereby excluding the attacks and decays of the notes) using a length of approximately 8000 samples (at a CD quality sampling rate of 44100 samples per second). This means that each sample had a length of approximately a fifth of a second.

In the most general sense, musical instruments can be considered to consist of two parts: a sound producing excitation mechanism, and a resonating body that amplifies and filters the output of the exciter. The two parts are usually considered to be a coupled system, such as the strings and body of a guitar. Both the clarinet and the saxophone are single-reed instruments, meaning that the exciter consists of a single piece of cane reed. However, despite having some common ancestry, there are some important differences between the two. Primarily, these are that the clarinet is made of wood and is considered to be a cylindrical bore instrument, while the saxophone is metallic and is primarily cone shaped. The many similarities between the two instruments make them good candidates for comparison.

Under certain conditions, a wind instrument can be made to emphasize sets of disparate harmonics that do not occur when played normally. This is known as a multiphonic tone, and is sometimes referred to as "playing two tones at the same time" by woodwind players. Multiphonics are an advanced playing technique, requiring great skill and a knowledge of special instrument fingerings. Because they alter the normal sound-producing mechanisms of the instrument, as well as requiring large amounts of breath pressure, these notes have been considered good candidates for previous studies of dynamic and chaotic behavior in musical instruments.

We also wanted to see how some of the reconstruction techniques would work on a synthetic clarinet tone. The synthesis method used for this project is the physical modeling using the STK toolkit [2]. Unlike other more common synthesis systems such as wave playback or frequency modulation synthesis, physical models employ waveguide delays to simulate the internal acoustics of an instrument. One of the advantages to using these models for this kind of work is that the individual parameters of the sound production mechanism such as breath pressure or reed stiffness (in the case of the clarinet) can be varied individually. Most of these parameters are coupled together in such a way that altering one will affect the other. For instance, a stiffer reed on a clarinet requires greater breath pressure for any kind of oscillation to occur. In user-based listening tests, physical models have been shown to have a high degree of perceptual similarity to real sounds. Nevertheless, expert listeners can usually hear the differences, and we wanted to know if this would also be visible under dynamic analysis conditions.

4 RESULTS

4.1 Attractor Reconstruction

4.1.1 2D Attractor Reconstruction

The need for choosing an appropriate lag time is illustrated if we try using a simple lag of 1 time step with our clarinet data in a 2-D embedding. This results in a shape very close to a diagonal line, x=y (Fig. 1).

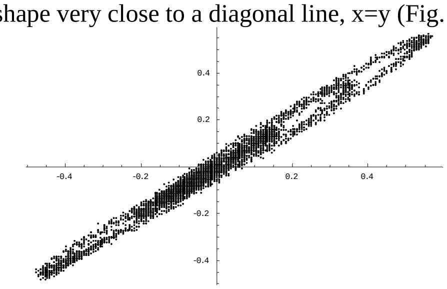

Figure 1: Clarinet, Lag 1

The problem is that due to the short real time interval between samples, any data point is quite highly correlated with the following point. Choosing a lag that sets the autocorrelation function to 0 results in a better spreading-out of points across phase space (Fig. 2).

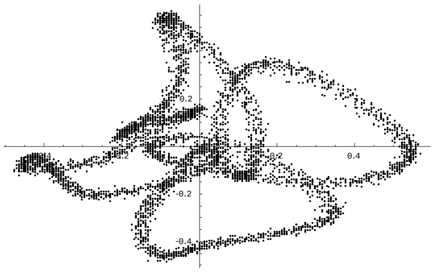

Figure 2: Clarinet, Lag 46

Note how the 2D reconstruction does not provide enough room for trajectories, as they still apparently intersect. The same thing occurs with the saxophone data (Fig. 3).

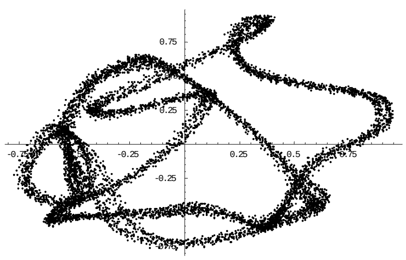

Figure 3: Sax, Lag 16

The synthesized clarinet, however, has a quite simple structure in a 2D embedding (Fig. 4).

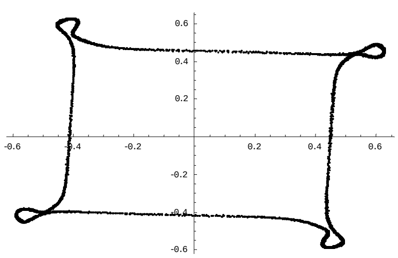

Figure 4: Synth Clarinet, Lag 33

4.1.2 3D Attractor Reconstruction

We can try 3D reconstructions to allow trajectories more room. Indeed, these pictures (Fig. 5) look better, but it still looks like there may be some self-intersection. Note how the general shape of the data for each sound sample looks like a torus that has been stretched around, possibly through higher dimensions.

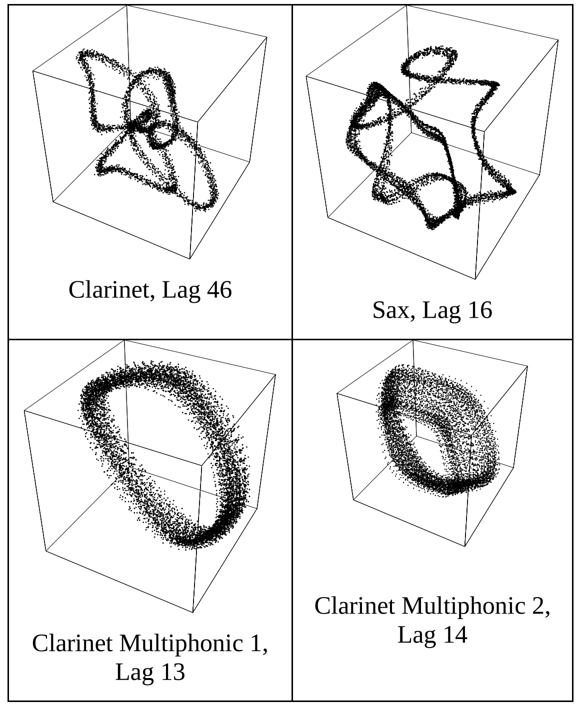

Figure 5: 3D Attractor Reconstructions

4.2 Autocorrelation & Largest Lyapunov Exponents

The results of these two computations are summarized below for each data set (Table 1).

Data	Largest Exponent	Time Lag
Clarinet	-0.049	46
Synth. Clarinet	-0.10	33
Sax	-0.14	16
Clarinet Multiphonic 1	-0.17	13
Clarinet Multiphonic 2	-0.17	14
Clarinet Multiphonic 3	-0.16	13

Table 1: Lyapunov Exponents and Time Lags

Interestingly, the clarinet sound, which turns out to have the highest-dimensional attractor in later analysis, also has the largest Lyapunov exponent – it is closer to having a chaotic attractor than the other sounds. Indeed, [1] references a clarinet model that predicts a period-doubling route to chaos as breath pressure is varied as a parameter. However, they claim that real clarinets do not give evidence of a chaotic regime. We were surprised that the multiphonics had the smallest exponents, because we expected less speedy convergence to the attractor due to the fuzzy quality of the 3D time lag plots. In any case, as mentioned above, all the largest exponents are negative, so although we were on the lookout for chaotic attractors, we find no evidence of such in these data sets.

Lags were determined by selecting the first lag time close to the autocorrelation function. Sample autocorrelation plots follow (Fig. 6-8).

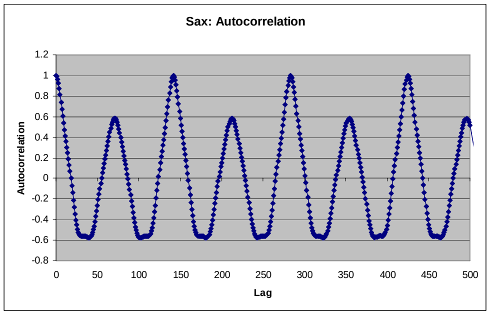

Figure 6: Sax autocorrelation

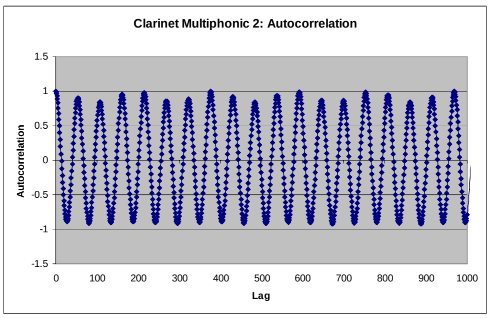

Figure 7: Multiphonic autocorrelation

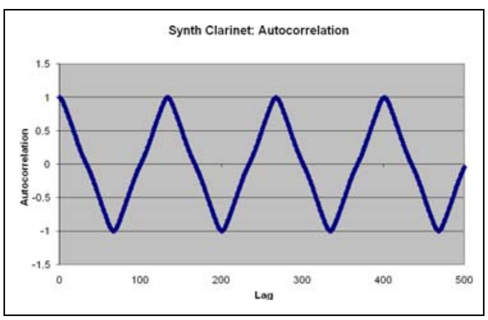

Figure 8: Synth clarinet autocorrelation

4.3 Prediction

Prediction error results are summarized in Figures 9-12. Generally, 3 or 4 dimensions seemed sufficient for low error, although the clarinet is slightly better at 7 dimensions (Fig. 9).

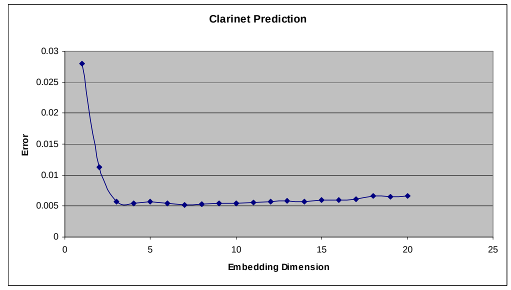

Figure 9: Clarinet prediction error

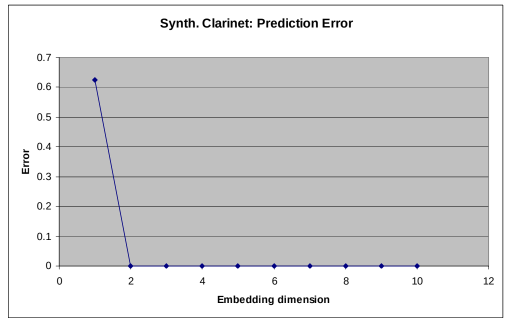

Figure 10: Clarinet prediction error

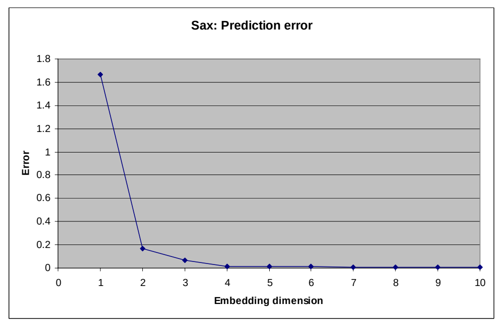

Figure 11: Clarinet prediction error

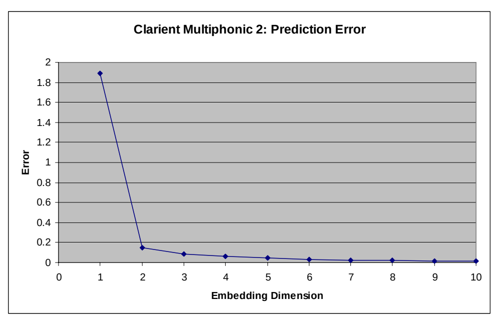

Figure 12: Multiphonic prediction error

4.4 Synthesis

For each data set, we used our simple nearest-neighbor predictor method to resynthesize a segment of audio. We generated more output data than we had as input in each case; this is an interesting advantage of this method over traditional wavetable synthesis, which requires the audio engineer to choose a segment of sound to loop and force the end point of the loop to match up well with the initial point. In our method the generated audio does not necessarily loop periodically. This data-implicit synthesis is less sophisticated than the neural network model in [7], but is easy to implement. (In [7] a neural net is trained over the data and then used to generate predictions in much the same way as our method, except that the neural net is used instead of the nearest neighbor). We listened to the results of the resynthesis over a range of embedding dimensions for the each data set. For the clarinet data, we also tried a different time lag (13 instead of 46), to test the rule of selecting an autocorrelation of 1/e (instead of 0). However, the sounds produced were noticeably lower quality.

In general, using an embedding dimension of 2 resulted in a very noisy, metallic, distorted sound. Using 3 or 4 dimensions improved quality greatly, although the sound was still inferior to the original audio; there were still extra high-frequency, noisy elements present. We also tried using more than one nearest neighbor for prediction, but averaging predictions of several neighbors resulted in terrible sound. There were obvious patterns in the raw data; many predictions would be made at several particular amplitude values, so the data would be concentrated in horizontal lines on a time vs. amplitude plot.

The synthesized clarinet resulted in the best resynthesis, as might be suspected due to the simplicity of its 2D attractor. Audio files of the synthesis experiments are available online at `http://www.cs.indiana.edu/~epnichol/q580/synthesis/`.

4.5 False NN Embedding dimension

Results for the false nearest neighbor algorithm were inconclusive, as demonstrated in Figure 13.

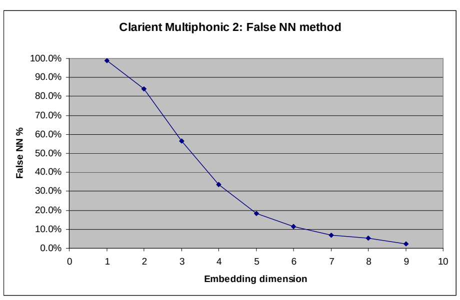

Figure 13: Multiphonic False NN Method

4.6 Correlation Sum

Using each set of log correlation sum, a region of nearly linear behavior was selected and slopes computed. These slopes are shown on graph for each data set giving slope (correlation dimension) vs. embedding dimension. The synthesized clarinet data was the simplest to interpret, with correlation dimension of approx. 0.4. This low value helps reinforce our view that the synthesized data is simpler than the other data sets, which each have higher correlation dimension. The saxophone data, for instance, has a correlation dimension near 1.2. The recorded clarinet data has the largest dimension of any data considered: the dimension is 2.5 at an embedding dimension of 7, and seems to still be increasing at that point. This may indicate that a higher-dimensional attractor is needed to explain the clarinet data; it is also likely that additional noise in that recording resulted in the higher dimensionality.

Finally, recall our earlier discussion of the inharmonicity of piano strings vibrating. The frequencies produced are not related by simple rational numbers and result in something like a 2-torus attractor. A clarinet multiphonic also has this property of producing multiple inharmonic tones at once. Thus, the dimensionality calculation of around 1.8 (and still increasing on this plot) is not surprising. The 3D embedding of this data looks like a quite fuzzy 2-torus; we suspect that this spread-out character is related to the characteristic timbre of the multiphonic.

If we apply the Takens embedding theorem with correlation dimension as our measure of attractor dimension, we can get an upper bound on the necessary embedding dimension for each data set. Table 2 summarizes these results, along with the dimension suggested by our simple prediction experiment graphs above (Figs. 9-12).

Data	**Correlation dimension**	**Predictor dimension**	**Max dim.**
Synth. Clarinet	0.4	2	3
Sax	1.2	4	4
Clarinet #2 Multiphonic	>1.8	4 (improves through 7)	4
Clarinet	>2.5	3-4, or 7	7

Table 2. Attractor dimension measures.

Our results seem consistent; in each case the dimension suggested by the simple predictor method falls between the expected minimum (correlation dimension) and maximum (Takens-theorem dimension). Figures 14-21 display the graphs used to approximate correlation dimensions.

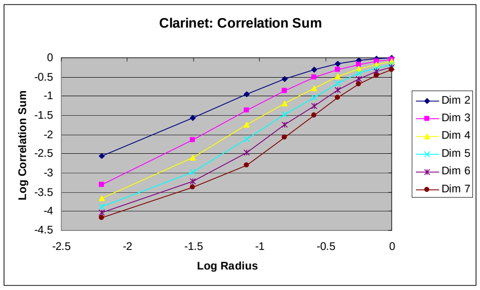

Figure 14: Clarinet correlation sum

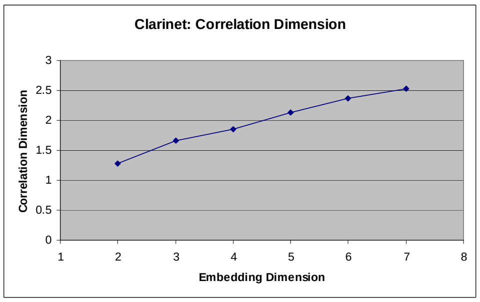

Figure 15: Clarinet correlation dimension

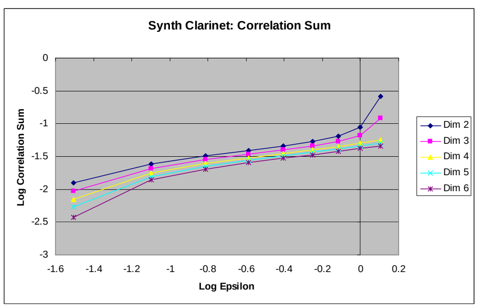

Figure 16: Synth clarinet correlation sum

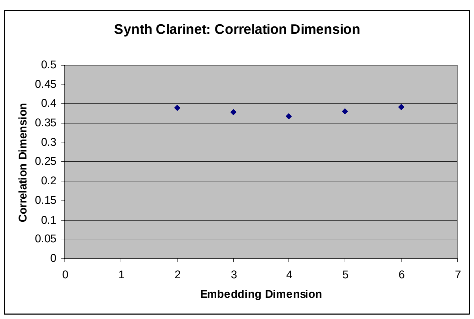

Figure 17: Synth clarinet correlation dimension

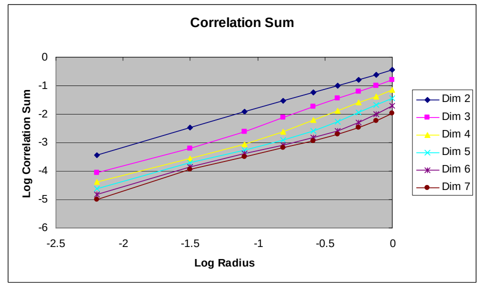

Figure 18: Sax correlation sum

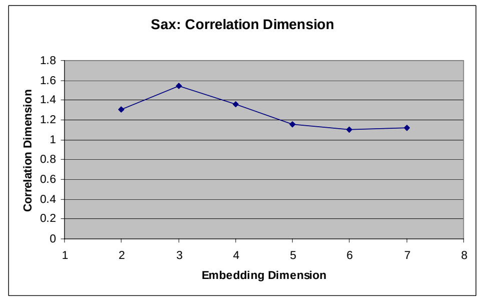

Figure 19: Sax correlation dimension

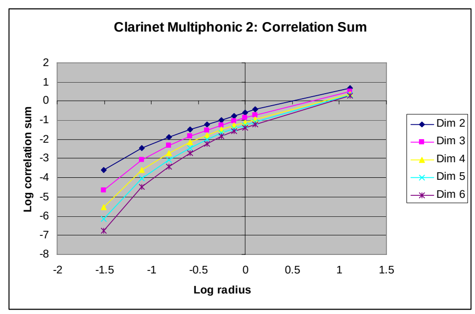

Figure 20: Clarinet multiphonic correlation sum

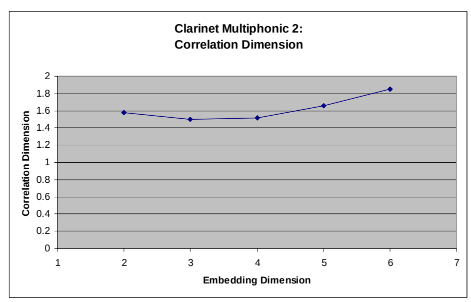

Figure 21: Clarinet multiphonic correlation dimension

4.7 LK Period Histogram

Sample period histograms for the LK method are shown below (Figs. 22-25). We were able to consistently use the maximum peak from these graphs to calculate the fundamental frequency of each pitch performed (frequency = 44.1kHz / period of peak). Note the simpler-looking distributions given for the clarinet multiphonic as well as the synthesized clarinet. For the multiphonic data, the two smaller peaks to the left of the largest peak probably correspond to inharmonic frequencies. The other histograms show a much more complex structure; many larger periods appear, corresponding to lower-frequency components to the sound. These may result from things such as low-frequency background noise and lower-frequency musical components such as vibrato in the recorded signals.

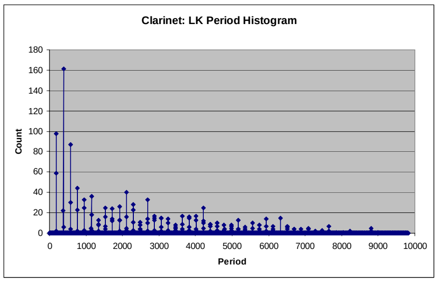

Figure 22: Clarinet LK histogram

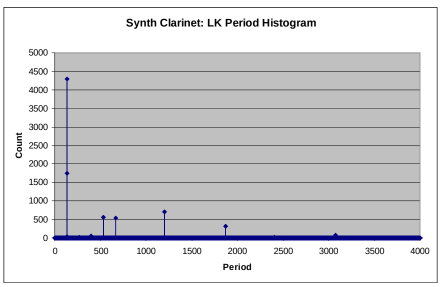

Figure 23: Synth clarinet LK histogram

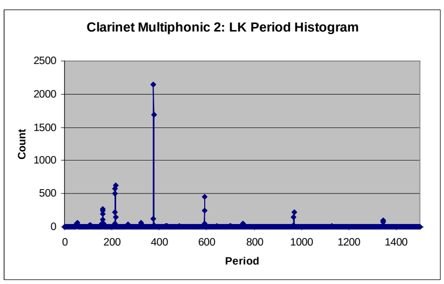

Figure 24: Multiphonic LK histogram

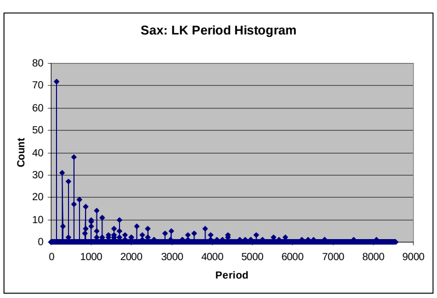

Figure 25: Sax LK histogram

5 APPLICATIONS

Besides being a novel synthesis method with interesting research potential on its own, there are several areas where this research may be of specific interest to audio engineers in general. Many synthesis methods, such as sample-based "wavetable" synthesis suffer from what Richard Moore has called the "dumb machine" effect [5]. This is when a sound has a repetitive or redundant quality that is immediately perceptible as being artificial. In a real instrument, each note differs slightly in small ways that avoid this repetitious quality, and yet are still similar enough to be perceived as to be perceptually grouped within the same instrument. For instance, everyone knows what a saxophone sounds like, but there are many different saxophones in the world of varying size, timbral color, and even quality, that are not perceptually captured by a single sampled waveform, or even a group of samples. However, much of chaos theory is concerned with exactly the question of the association between surface variation and underlying pattern. We believe chaos-based synthesis techniques have the potential to synthesize sounds that contain the small differences necessary to reduce or eliminate the "artificialness" of the sound, yet still maintain an adequate enough degree of similarity between instances that most listeners will have no difficulty identifying these as being from the same natural, sound-generating source.

A second, related area is in the area of compression techniques. As a mathematical technique, attractors are designed to clarify the repetitive parts of a chaotic data set. By identifying the aspects that are cyclical, and providing a method of recreating the parts that are not, a data set can be effectively summarized using less information than contained in the original. This is essentially the basis for a compression technique, and we are currently exploring this avenue of research.

6 CONCLUSION

We have presented a collection of methods useful in an exploration of attractors in audio signals. All the techniques presented here were quite straightforward to implement and did not require too much computation time (the K-nearest neighbors problems took the most time, but can be implemented more efficiently with standard algorithms). We would have liked to implement the Lyapunov exponent calculation for more than the largest exponent, but this is more difficult and, it turns out, unnecessary for the data we used.

We used autocorrelation to estimate optimal lag time. Although other techniques such as mutual information [9] may give better results, our method seemed, empirically, to work quite well.

In general, we found the correlation sum to be a much more useful indicator of dimensionality than the false nearest neighbor algorithm. The correlation sum included an algorithmic way to vary the radius, while the false nearest neighbor method required us to make a particular choice, and the results seemed to demonstrate large variations with different choices of radius. The prediction-based dimension also turned out to give consistent predictions of attractor dimension.

Although we initially expected to find chaos in complex audio data, such as in clarinet multiphonics, we found all our data to have negative Lyapunov exponents. It may be the case that *musical* sounds generally exhibit attractors, but not necessarily chaotic attractors.

Finally, in addition to characterizing attractor and phase-space (embedding) dimension from time-series data, we were able to produce a sound synthesizer based on the calculated embedding parameters, with a much simpler algorithm than the neural network in [7].

While the current method is primarily limited to the steady-state portions of a sound, there is potential for interesting future work on resynthesis based on this method, and we are currently working towards improving upon the results presented here.

REFERENCES

[1] Bernardi, A., Bugna, G., De Poli, G. (1997). “Musical signal analysis with chaos”, in Curtis Roads et al., eds., *Musical Signal Processing* (Lisse, Netherlands: Svets and Zeitlinger) pp. 187–220.

[2] Cook, P. R. and Scavone, G. (1999). "The Synthesis ToolKit (STK)", *Proceedings of the International Computer Music Conference, Beijing.*

[3] Kaplan, D. and Glass, L. (1995). *Understanding Nonlinear Dynamics*. New York: Springer-Verlag.

[4] Kennel, M., Brown, R., and Abarbanel, H. “Determining embedding dimension for phase-space reconstruction using a geometrical construction”. *Phys. Rev. A. 45*(6), 15 March 1992. pp. 3403-3411.

[5] Moore, Richard. (1990). *Elements of Computer Music*. Englewood Cliffs, N.J.: Prentice Hall. p. 279.

[6] Peitgen, Jürgens, and Saupe. (1992). *Chaos and Fractals: New Frontiers of Science.* New York: Springer-Verlag.

[7] Röbel, A. (1999). “Morphing Sound Attractors”, *Proc. of the 3rd. World Multiconference on Systemics, Cybernetics and Informatics (SCI'99) and the 5th. Int'l Conference on Information Systems Analysis and Synthesis (ISAS'99), Vol VI.* Skokie, Illinois: International Institute of Informatics and Systemics, pp. 340–347.

[8] Röbel, A. “Synthesizing Natural Sounds using Dynamic Models of Sound Attractors.” *Computer Music Journal*, *25*(2), pp. 46-61, Summer 2001.

[9] Williams, G. (1997). *Chaos Theory Tamed.* Washington, D.C.: Joseph Henry Press.

[10] Ying-Chang, L., Lia, Z. and Ye, N. “Nonlinear Time Series Analysis” in *The Handbook of Data Mining*, Nong Ye, Ed. 2003.