Data-driven Modeling of Acoustical Instruments

Bernd Schoner, Charles Cooper, Christopher Douglas, Neil Gershenfeld

Physics and Media Group, MIT Media Lab, Cambridge, MA 02139

Tel: (617)253-6581, Fax: (617)258-7168

schoner@media.mit.edu

(December 9, 1998)

Abstract

We present a framework for the analysis and synthesis of acoustical instruments based on data-driven probabilistic inference modeling. Audio time series and boundary conditions of a played instrument are recorded and the non-linear mapping from the control data into the audio space is inferred using the general inference framework of Cluster-Weighted Modeling. The resulting model is used for real-time synthesis of audio sequences from new input data.

I. INTRODUCTION

Sampling of acoustical instruments [Massie, 1998] and synthesis based on detailed first-principles physical modeling [Smith, 1992] have been two particularly successful musical synthesis techniques. The sampling approach typically results in high sound quality, but has no notion of the instrument as a dynamic system with variable control. The physical modeling approach retains this dynamic control but results in intractably large models when all the physical degrees of freedom are considered. The search for the right combination of model parameters is difficult and there is no systematic way to ascertain and incorporate subtle differences between instruments of the same class, such as two master violins. We present a new synthesis method that is conceptually intermediate between these two traditional techniques, in that it infers the physical behavior of the instrument from observation of its global performance.

Dynamical systems theory shows that we can reconstruct a state space of a physical system that is homeomorphic to the actual state space using the input and output observables of the system along with their time lags [Takens, 1981,Casdagli, 1992]. The reconstructed space captures the dynamics of the entire system. At the same time, its dimensionality may be chosen to correspond to the number of effective rather than actual degrees of freedom of the system. In the case of dissipative systems, such as musical instruments, this effective dimensionality can be considerably lower than the physical one. We combine this result with adequate signal processing and sensing technology to build a modeling and synthesis framework that introduces new capabilities and control flexibility into accepted and mature synthesis techniques.

Using the violin as our test instrument, we developed unobtrusive sensors that track the position of the bow relative to the violin, the pressure of the forefinger on the bow, and the position of the finger on the fingerboard. In a training session, we record control input data from these sensors along with the violin's audio output. These signals serve as training data for the inference engine, Cluster-Weighted Modeling, which learns the non-linear mapping between the control inputs and the target audio output. The resulting model can predict audio data based on new control data. A violinist plays the interface device (which could be a silent violin), and the sensors now drive the computer model to produce the sound of the original violin.

We have developed Cluster-Weighted Modeling (CWM) as a general prediction and characterization inference engine that naturally extends beyond linear inference and signal processing practice into a powerful non-linear framework that handles non-Gaussianity, non-stationarity, and discontinuity in the data. CWM retains the functionality of conventional neural networks, while addressing many of their limitations.

II. DATA COLLECTION

Several sensors capture the gestural input of the violin player. We measure the violin bow position with a capacitive coupling technique that detects displacement current in a resistive antenna [Paradiso and Gershenfeld, 1997] and infer a bow velocity estimate by differentiation and Kalman filtering. The distance between bow and violin bridge is determined by the same technique. Bow pressure is inferred from the force exerted by the player's index finger on a force-sensitive resistor mounted on the bow.

The finger-position sensor consists of a thin strip of stainless-steel ribbon attached to the finger board. A low frequency alternating current passes through the violin string and is divided according to the distance between the contact point and the two ends of the ribbon. We infer the position of the player's finger from the difference between the two currents and use the sum of the currents to determine whether the string is in contact with the finger board. A microphone placed close to the violin detects the acoustic sound pressure and a dynamic pickup measures the string vibration signal. The pickup consists of a permanent magnet mounted underneath the strings and an amplifier that detects the voltage induced in the string.

During data recording sessions, we simultaneously record sensor and audio data (see Fig. 1). For performance, the violinist plays the instrument with the strings covered by a shield that prevents bow-string contact. The sensor signals are again fed into the computer, and now a real-time program predicts sound parameters and synthesizes audio output corresponding to the player's performance data.

[INSERT FIGURE 1 ABOUT HERE]

III. MODELING SEQUENCE

The model building and prediction sequence is broken into the following steps: (1) transform the output data representation from time series audio samples to spectral data,

(2) prepare the state space representation from input and output data series, (3) build an input-output prediction model using the cluster-weighted algorithm, (4) use this model to predict output spectral data based on new input data, and (5) synthesize the audio stream from the predicted spectral sequence.

A. Data Analysis and Representation

Our first attempts at using embedding synthesis to model driven input-output systems were entirely time domain-based. While this approach is close to the techniques suggested by dynamical systems theory [Casdagli, 1992], it suffered from instability due to the difference in characteristic time scales between control inputs and signal outputs. In addition, the time domain approach retains some perceptually irrelevant features and unpredictable information. For example, although signal phase is perceivable under special circumstances, the violin player is not actively controlling the phase of the signal components. The player is, however, actively shaping the spectral energy characteristics of the audio signal. We therefore choose to predict in the spectral domain [McAulay and Quatieri, 1985,Serra and Smith, 1990] and hence model the process rather than a particular realization of the process.

We decompose the measured audio signal into spectral frames in such a way that each audio frame corresponds to a measured set of input variables $\mathbf{x}'$. Each frame is obtained from a Short Term Fourier Transform (STFT) applied to audio samples weighted by a Hamming window. Only the information corresponding to harmonic components of the signal (peaks in the spectrum) is retained. The amplitude of each partial is taken to be proportional to the magnitude of the STFT, and we compute a precise estimate of the partial frequency from the phase difference of STFTs applied to two windows shifted by a single sample [McAulay and Quatieri, 1986,Brown and Puckette, 1993]. After analysis, the training data is reduced to the set of input-output points $\{\mathbf{y}_n, \mathbf{x}'_n\}_{n=1}^{N}$, where the index n refers to consecutive frames, $\mathbf{x}'$ refers to the vector of inputs consisting of bow velocity, pressure, finger position, and bow bridge position, and $\mathbf{y}$ refers to the vector of partials with each partial represented by a pair describing frequency and amplitude.

We begin the modeling task by evaluating the length of the instrument's "memory" and finding the input signals that best represent the influence of the past on the current output. To this end we add selected time-lagged components of $\mathbf{x}'$ to the input vector and obtain the augmented input vector $\mathbf{x}$. In particular, we attempt to augment the input data set so that the input-output mapping becomes a single-valued function. The performance of a particular feature vector is evaluated by cross-validation, that is, by prediction and resynthesis from input data that was not used during model building.

The input-output pairs $\{\mathbf{x}_n, \mathbf{y}_n\}$ are used to train the inference algorithm as described in the next section. The resulting model generates the estimated output $\hat{\mathbf{y}}$ given a new input vector $\mathbf{x}$. From the spectral vector $\hat{\mathbf{y}}$ we reconstruct a time domain waveform by linearly interpolating frequencies and amplitudes of the partials between frames. The sinusoidal components corresponding to the partials are summed into a single audio signal [Serra and Smith, 1990].

B. Cluster-Weighted Modeling

Cluster-Weighted Modeling (CWM) is an input-output inference framework based on probability density estimation of a joint set of input features and output target data. It is similar to hierarchical mixture-of-experts type architectures [Jordan and Jacobs, 1994] and can be interpreted as a flexible and transparent technique for approximating an arbitrary function. During training, clusters automatically "go to where the data is" and approximate subsets of the data space according to a smooth domain of influence (Fig.2). Globally, the influence of the different clusters is weighted by Gaussian basis terms, while locally, each cluster represents a simple model such as a linear regression function. Thus, previous results from linear systems theory, linear time series analysis and traditional musical synthesis are applied within the broader context of a globally non-linear model.

[INSERT FIGURE 2 ABOUT HERE]

After preprocessing the experimental measurements we obtain the set of training data $\{\mathbf{y}_n, \mathbf{x}_n\}_{n=1}^{N}$, where $\mathbf{x}$ refers to the feature input vector and $\mathbf{y}$ refers to the corresponding target output vector. We infer the joint probability density of feature and target vector $p(\mathbf{y}, \mathbf{x})$, which lets us derive conditional quantities such as the expected value of $\mathbf{y}$ given $\mathbf{x}$, $\langle \mathbf{y}|\mathbf{x}\rangle$, and the expected covariance matrix of $\mathbf{y}$ given $\mathbf{x}$, $\langle \mathbf{C}_{yy}|\mathbf{x}\rangle$. The value $\langle \mathbf{y}|\mathbf{x}\rangle$ serves as prediction of the target value $\mathbf{y}$ and $\langle \mathbf{C}_{yy}|\mathbf{x}\rangle$ estimates the prediction error [Gershenfeld et al., 1998].

The joint density $p(\mathbf{x}, \mathbf{y})$ is expanded in clusters c_m, each of which contains an input domain of influence, a local model, and an output distribution:

$$\begin{aligned} p(\mathbf{y}, \mathbf{x}) &= \sum_{m=1}^{M} p(\mathbf{y}, \mathbf{x}, c_m) && (1) \\ &= \sum_{m=1}^{M} p(\mathbf{y}|\mathbf{x}, c_m)\ p(\mathbf{x}|c_m)\ p(c_m) \quad . \end{aligned}$$

The probability functions $p(\mathbf{y}|\mathbf{x}, c_m)$ and $p(\mathbf{x}|c_m)$ are taken to be of Gaussian form so that $p(\mathbf{x}|c_m) = N(\mu_m, \mathbf{C}_m)$ and $p(\mathbf{y}|\mathbf{x}, c_m) = N(\mathbf{f}(\mathbf{x}, \beta_m), \mathbf{C}_{y,m})$, where $N(\mu, \mathbf{C})$ stands for the multi-dimensional Gaussian distribution with mean vector μ and covariance matrix $\mathbf{C}$. The function $\mathbf{f}(\mathbf{x}, \beta_m)$ with unknown parameters β_m should be taken to be a generalized linear model [Gershenfeld, 1998], for example a polynomial model.

The complexity of the local model is traded off against the complexity of the global architecture. In the case of polynomial expansion, there are two extreme cases that illustrate this trade-off. We may use locally constant models in connection with a large number of clusters, in which case the predictive power comes only from the number of Gaussian kernels. Alternatively we may decide to use a high-order polynomial model and a single kernel, in which case the model reduces to a global polynomial model. In this particular application we work with local linear models

$$f(\mathbf{x}, \beta_m) = \beta_{0,m} + \sum_{d=1}^{D} \beta_{d,m} x_d \quad , \tag{2}$$

where d refers to an input dimension and D to the total number of input dimensions.

Given the density estimate, we can analytically infer a conditional forecast

$$\begin{aligned} \langle \mathbf{y}|\mathbf{x}\rangle &= \int \mathbf{y} p(\mathbf{y}|\mathbf{x}) d\mathbf{y} \\ &= \int \mathbf{y} \frac{p(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})} d\mathbf{y} \\ &= \frac{\sum_{m=1}^{M} \int \mathbf{y}\ p(\mathbf{y}|\mathbf{x}, c_m)\ d\mathbf{y}\ p(\mathbf{x}|c_m)\ p(c_m)}{\sum_{m=1}^{M} p(\mathbf{x}|c_m)\ p(c_m)} \\ &= \frac{\sum_{m=1}^{M} \mathbf{f}(\mathbf{x}, \beta_m)\ p(\mathbf{x}|c_m)\ p(c_m)}{\sum_{m=1}^{M} p(\mathbf{x}|c_m)\ p(c_m)} \quad . \end{aligned} \tag{3}$$

as well as a conditional error forecast,

$$\langle \mathbf{C}_{yy}|\mathbf{x}\rangle = \frac{\sum_{m=1}^{M} [\mathbf{C}_{y,m} + \mathbf{f}(\mathbf{x}, \beta_m) \cdot \mathbf{f}(\mathbf{x}, \beta_m)^T]\ p(\mathbf{x}|c_m)\ p(c_m)}{\sum_{m=1}^{M} p(\mathbf{x}|c_m)\ p(c_m)} - \langle \mathbf{y}|\mathbf{x}\rangle^2 \quad . \tag{4}$$

The choice of the number of clusters M controls under- versus over-fitting. The model should be given enough clusters to model the predictable data, but should not become so complex that it predicts the noise and other non-generalizable features. The optimal M can be determined by cross-validation with respect to a mathematical error measure such as the square error or with respect to perceptual performance.

We find the model parameters using a variant of the Expectation-Maximization (EM) algorithm [Dempster et al., 1977,Jordan and Jacobs, 1994]: the unconditioned cluster probabilities $p(c_m)$, cluster locations μ_m, and covariances $\mathbf{C}_m$ are estimated in conventional EM updates. We then use pseudo-inverses of the cluster weighted covariance matrices to update the local model parameters β_m. The EM algorithm finds the most likely cluster parameters by iterating between an expectation step and a maximization step.

E-step: Given a starting set of parameters, we compute the probability of a cluster given a data:

$$
\begin{aligned}
p(c_m|\mathbf{y},\mathbf{x}) &= \frac{p(\mathbf{y},\mathbf{x}|c_m)\ p(c_m)}{p(\mathbf{y},\mathbf{x})} \\
&= \frac{p(\mathbf{y},\mathbf{x}|c_m)\ p(c_m)}{\sum_{l=1}^{M} p(\mathbf{y},\mathbf{x}|c_l)\ p(c_l)} \quad ,
\end{aligned} \tag{5}
$$

where the sum over clusters in the denominator lets clusters interact and specialize in data they best explain.

M-step: Now we assume the current data distribution is correct and maximize the likelihood function by re-computing the cluster parameters. The new estimate for the unconditioned cluster probabilities becomes:

$$
\begin{aligned}
p(c_m) &= \int p(c_m|\mathbf{y},\mathbf{x})\ p(\mathbf{y},\mathbf{x})\ d\mathbf{y}\ d\mathbf{x} \\
&\approx \frac{1}{N}\sum_{n=1}^{N} p(c_m|\mathbf{y}_n,\mathbf{x}_n) \quad .
\end{aligned} \tag{6}
$$

The cluster-weighted expectation of any function $\theta(\mathbf{x})$ is defined as

$$
\begin{aligned}
\langle\theta(\mathbf{x})\rangle_m &\equiv \int \theta(\mathbf{x})\ p(\mathbf{x}|c_m)\ d\mathbf{x} \\
&= \int \theta(\mathbf{x})\ p(\mathbf{y},\mathbf{x}|c_m)\ d\mathbf{y}\ d\mathbf{x} \\
&= \int \theta(\mathbf{x}) \frac{p(c_m|\mathbf{y},\mathbf{x})}{p(c_m)}\ p(\mathbf{y},\mathbf{x})\ d\mathbf{y}\ d\mathbf{x} \\
&\approx \frac{1}{N}\sum_{n=1}^{N} \theta(\mathbf{x}_n)\ \frac{p(c_m|\mathbf{y}_n,\mathbf{x}_n)}{p(c_m)} \\
&= \frac{\sum_{n=1}^{N} \theta(\mathbf{x}_n)\ p(c_m|\mathbf{y}_n,\mathbf{x}_n)}{\sum_{n=1}^{N} p(c_m|\mathbf{y}_n,\mathbf{x}_n)} \quad .
\end{aligned} \tag{7}
$$

This lets us update the cluster means and the cluster weighted covariance matrices :

$$
\begin{aligned}
\mu_m &= \langle\mathbf{x}\rangle_m \\
[\mathbf{C}_m]_{ij} &= \langle(x_i-\mu_i)(x_j-\mu_j)\rangle_m \quad .
\end{aligned} \tag{8}
$$

The derivation of the maximum likelihood solution for the model parameters yields

$$
\beta_m = \mathbf{B}_m^{-1} \cdot\ \mathbf{A}_m \quad , \tag{9}
$$

with $[\mathbf{B}_m]_{ij} = \langle f_i(\mathbf{x}, \beta_m) \cdot f_j(\mathbf{x}, \beta_m) \rangle_m$ and $[\mathbf{A}_m]_{ij} = \langle y_i \cdot f_j(\mathbf{x}, \beta_m) \rangle_m$.

Finally the output covariance matrices associated with each model are estimated,

$$\mathbf{C}_{y,m} = \langle [\mathbf{y} - \mathbf{f}(\mathbf{x}, \beta_m)] \cdot [\mathbf{y} - \mathbf{f}(\mathbf{x}, \beta_m)]^T \rangle_m \quad . \tag{10}$$

We iterate between the E- and the M-step until the overall likelihood of the data, as defined by the product of all data likelihoods (Equ.1) does not increase further.

IV. EXPERIMENTAL RESULTS

We collected approximately 30 minutes of input-output violin data, both single notes and scales with various bow strokes (Fig.1). We then built models based on subsets of this data, and used them for off-line and on-line synthesis.

The model performs very well on limited subsets of the overall training data. It is particularly robust at representing pitch and amplitude fluctuations caused by vibrato. Figure 3 illustrates that CWM can reproduce the spectral characteristics of a segment of violin sound. Three sound examples (`http://www.media.mit.edu/physics/publications/papers/cwm/`) demonstrate the results of audio resynthesis both in- and out-of-sample. Only the first half of each original sound file was used for training. We were able to build a real-time system running on Windows NT using a Pentium II 300 MHz. The system responds well to dynamic control changes, but considerable latency is caused by the operating system, data acquisition board and sound card.

[INSERT FIGURE 3 ABOUT HERE]

Models generalize better with respect to the input signals when they are trained to predict the string vibration signal rather than the sound pressure signal recorded by the microphone. The string vibration signal is considerably less complex, as it is not filtered by the violin body response. The final audio signal can be obtained by convolving the predicted string vibration signal with a measured impulse response of the acoustic violin [Cook and Truman, 1998].

V. CONCLUSIONS AND FUTURE WORK

We have shown how the general inference framework CWM can be used in a sensing and signal-processing system for musical synthesis. The approach is particularly appropriate for musical instruments that rely on continuous human control, such as the violin, since CWM naturally relates continuous input and output time series.

CWM overcomes many of the limitations of conventional inference techniques, yet it can only be as good as the audio representation. The current spectral representation does not provide model flexibility, makes strong assumptions about the nature of the physical device, and misses certain elements of the natural sound. Future work will therefore focus on improving the mutual interplay of representation and inference by embedding local filter and sample architectures and explicit constraints into the general CWM framework.

In this work we have shown how an approach intermediate between physical modeling and sampling combines features of traditional synthesis techniques to generate a model that provides both control flexibility and high fidelity to the original. More effort needs to be devoted to system integration and representation in order to progress from the current playable model to truly high quality instruments, but these preliminary results indicate the promise of “physics sampling”.

VI. ACKNOWLEDGMENTS

The authors would like to thank Romy Shioda, Sandy Choi and Teresa Marrin for playing the sensor violin, Edward Boyden for writing parts of the code used in this work, and Joe Paradiso for designing the violin bow. This work was made possible by the Media Lab's Things That Think consortium.

REFERENCES

BrownandPuckette,1993 Brown, J. C. and Puckette, M. S. (1993). A high resolution fundamental frequency determination based on phase changes of the fourier transform. *J. Acoust. Soc. Am.*, 94(2):662–667.

Casdagli,1992 Casdagli, M. (1992). A dynamical systems approach to modeling input-output systems. In Casdagli, M. and Eubank, S., editors, *Nonlinear Modeling and Forecasting*, Santa Fe Institute Studies in the Sciences of Complexity, pages 265–281, Redwood City. Addison-Wesley.

CookandTruman,1998 Cook, P. and Truman, D. (1998). A datatbase of measured musical instrument body radtioation and impulse rsponses, and computer applications for exploring and utilizing the measured filtered functions. In *Proceedings International Symposium on Musical Acoustics 1998*.

Dempsteret al.,1977 Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood From Incomplete Data via the EM Algorithm. *J. R. Statist. Soc. B*, 39:1–38.

Gershenfeld,1998 Gershenfeld, N. (1998). *The Nature of Mathematical Modeling*. Cambridge University Press, New York.

Gershenfeldet al.,1998 Gershenfeld, N., Schoner, B., and Metois, E. (1998). Cluster-weighted modeling for time series analysis. To appear in NATURE.

JordanandJacobs,1994 Jordan, M. and Jacobs, R. (1994). Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214.

Massie,1998 Massie, D. C. (1998). Wavetable sampling synthesis. In Kahrs, M.

and Brandenburg, K., editors, *Applications of Digital Signal Processing to Audio and Acoustics*, pages 311–341. Kluwer Academic Publishers.

McAulayandQuatieri,1985 McAulay, R. and Quatieri, T. (1985). Speech analysis/synthesis based on a sinusoidal representation. Technical Report 693, Massachusetts Institute of Technology / Lincoln Laboratory, Cambridge, MA.

McAulayandQuatieri,1986 McAulay, R. and Quatieri, T. (1986). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-34 No.4:744–754.

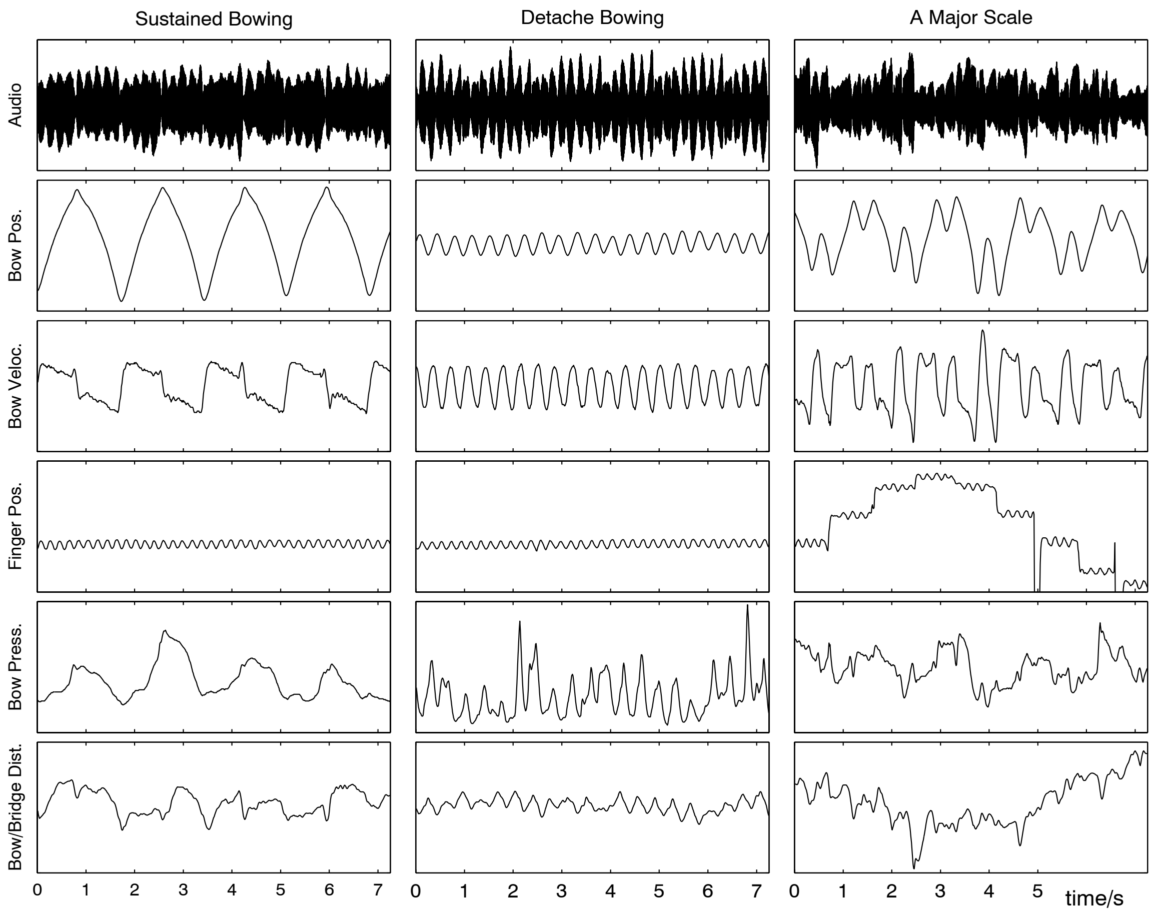
ParadisoandGershenfeld,1997 Paradiso, J. A. and Gershenfeld, N. (1997). Musical applications of electric field sensing. *Computer Music Journal*, 21(2):69–89.

SerraandSmith,1990 Serra, X. and Smith, J. O. (1990). Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24.

Smith,1992 Smith, J. O. (1992). Physical modeling using digital waveguides. *Computer Music Journal*, 6(4).

Takens,1981 Takens, F. (1981). Detecting strange attractors in turbulence. In Rand, D. and Young, L., editors, *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381, New York. Springer-Verlag.

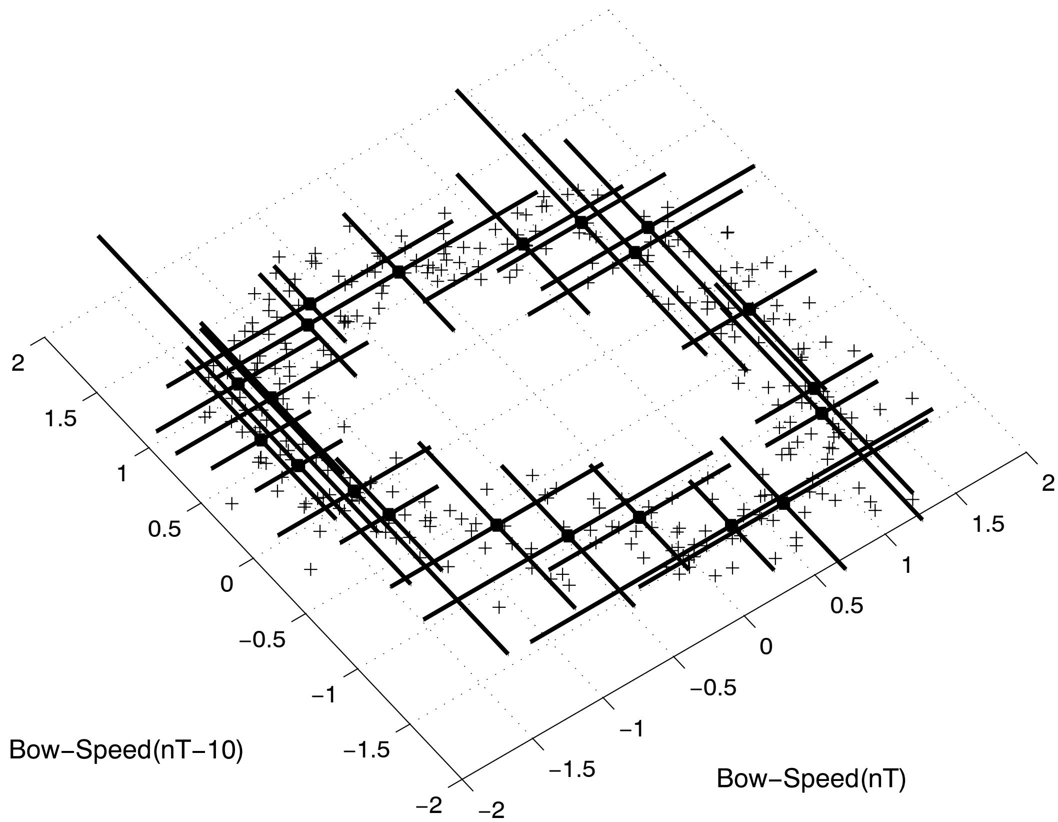
FIGURES



Sustained Bowing
Detache Bowing
A Major Scale
Audio
Bow Pos.
Bow Veloc.
Finger Pos.
Bow Press.
Bow/Bridge Dist.
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5
time/s

FIG. 1. Audio and input sensor data for various bowings and notes. *Left column:* E-natural, sustained bowing with strong vibrato. *Middle column:* E-natural, détaché bowing. *Right column:* A-major scale.

a)


Bow−Speed(nT−10)
Bow−Speed(nT)

b)

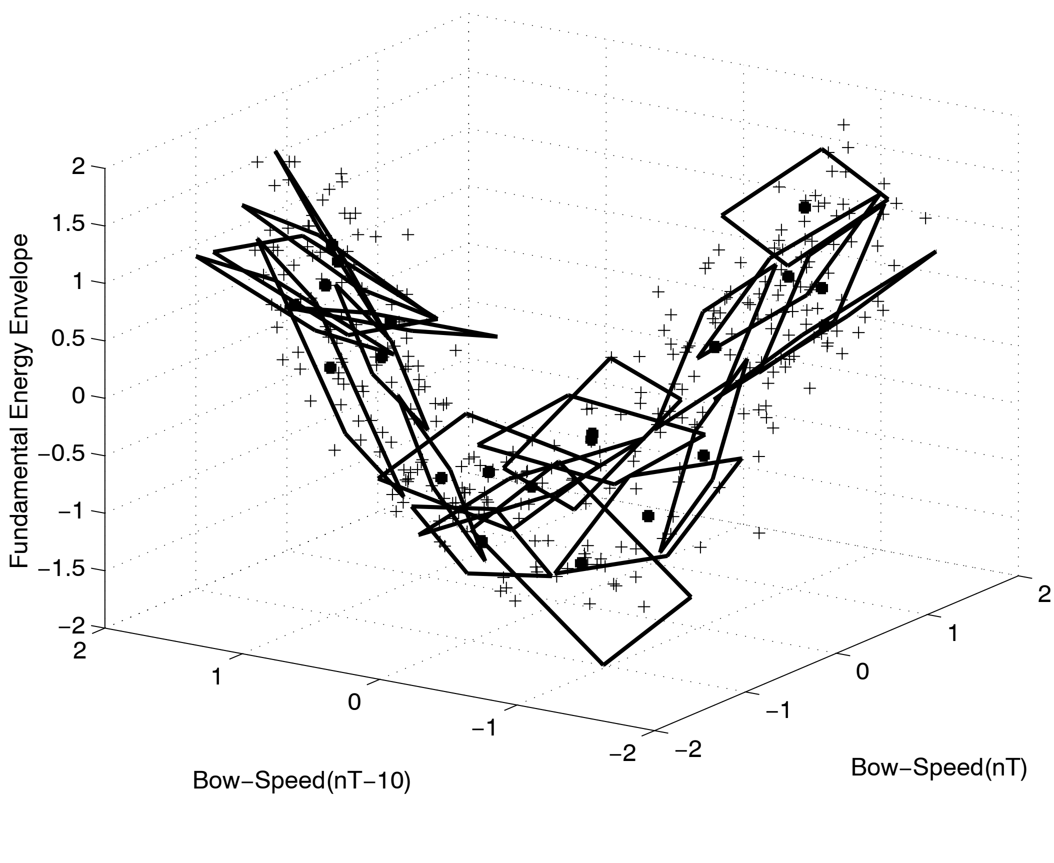

Fundamental Energy Envelope
Bow−Speed(nT−10)
Bow−Speed(nT)

FIG. 2. Data and clusters in the joint input-output space. *a:)* Vertical view of the input space. Clusters are represented by their centers and their domains of influence (variances). *b:)* Clusters in 3 dimensional input-output space with the rectangles representing the plane of the linear functions.

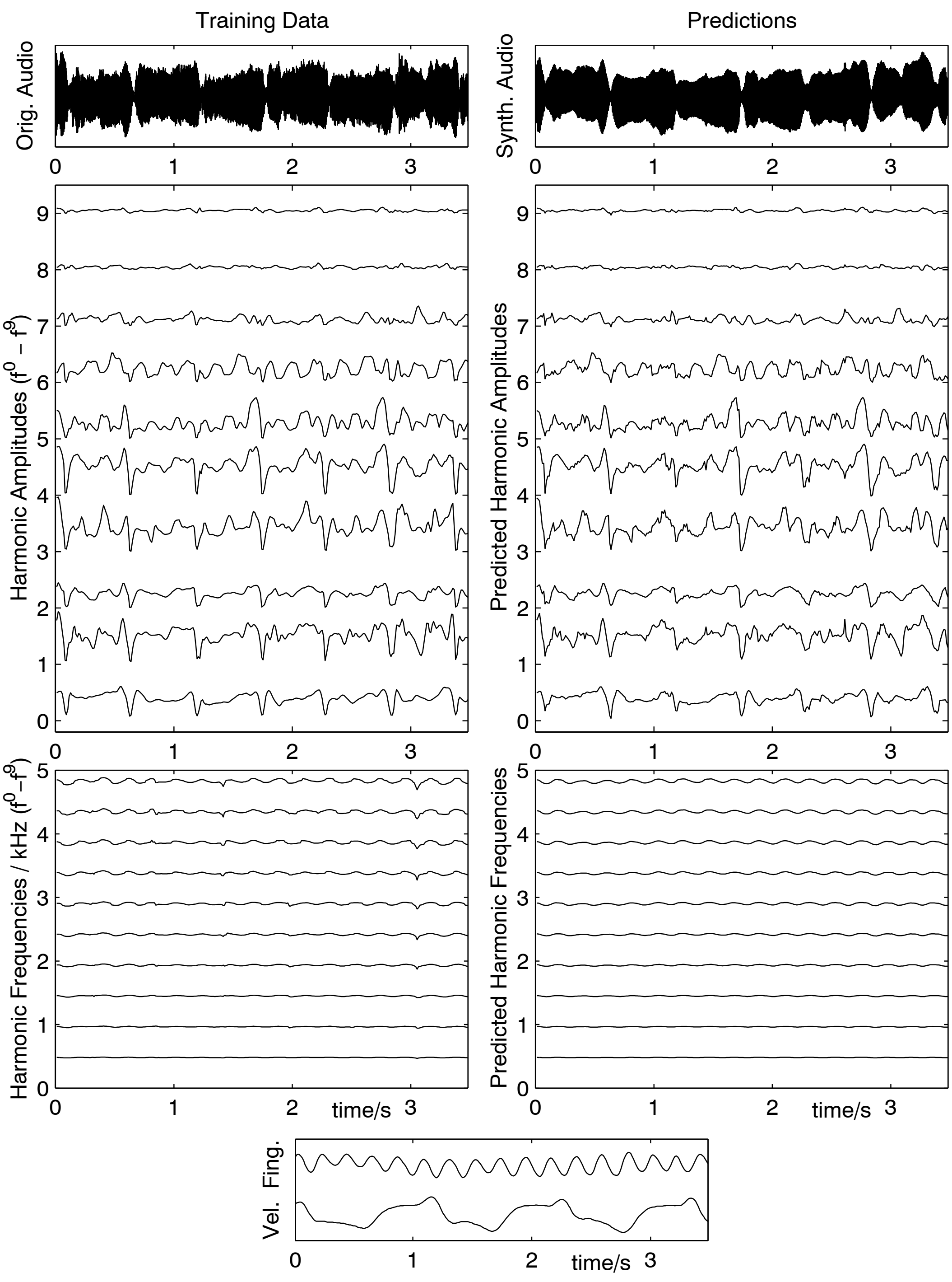

Training Data
Predictions
Orig. Audio
Synth. Audio
Harmonic Amplitudes (f^0 – f^9)
Predicted Harmonic Amplitudes
Harmonic Frequencies / kHz (f^0–f^9)
Predicted Harmonic Frequencies
time/s
time/s
Vel. Fing.
time/s

FIG. 3. Comparison of original and predicted violin time series data. *Bottom:* Input sensor measurements, showing the bow velocity (Vel.) and the player's finger position (Fing.). *Left:* The harmonic structure of the training data and the corresponding audio time series. *Right:* The predicted harmonic structure and the re-synthesized audio time series.

Bernd Schoner

MIT Media Lab, Physics and Media Group, 20 Ames St., Cambridge, MA 02139, USA

E-mail: schoner@media.mit.edu

http://www.media.mit.edu/ schoner/

Bernd Schoner was born in Germany in 1969. He received engineering diplomas (M.Sc.) in Electrical Engineering from Rheinische Wesfälische Technische Hochschule Aachen, Germany, and in Industrial Engineering from École Centrale de Paris, France, both in 1996. Since then he has been a Ph.D. candidate at the MIT Media Laboratory working with Prof. Neil Gershenfeld on the prediction and analysis of driven dynamical systems. His main research interests include machine learning, statistical inference, and the application of these techniques to problems in computer music and musical synthesis.

Charles Cooper

MIT Media Lab, Physics and Media Group, 20 Ames St., Cambridge, MA 02139, USA

E-mail: cmc@media.mit.edu

Chuck Cooper received the B.S. degree in electrical engineering from M.I.T. in 1967, and the M.S. degree in electrical and biomedical engineering from the University of Texas at Austin in 1968. After teaching undergraduate physics and electronics for three years, he became a research associate at the Harvard School of Public Health, where he developed medical computer software. In 1978, he founded a medical software company but covertly pursued his avocational interests in computer sound and psychoacoustics. After selling the company in 1996, Cooper joined the MIT Media Laboratory as a part-time Visiting Scientist. His interests include the challenge of generating sounds that seem “natural” rather than “electronic,” and the invention of hybrid acoustic-electronic musical instruments. He pursues these projects as the founder and sole employee of Plangent Systems Corporation.

Christopher Douglas

MIT Media Lab, Physics and Media Group, 20 Ames St., Cambridge, MA 02139, USA

E-mail:cdouglas@mit.edu

Christopher Douglas was born in Pasadena, California in 1980. He is now a senior at the Massachusetts Institute of Technology. His studies there include mathematics, machine learning, philosophy, comparative literature, and landscape history. Since 1997, he has been doing research on the design and application of machine learning algorithms with Prof. Neil Gershenfeld at the MIT Media Laboratory. He plans to pursue graduate work in either algebraic topology or low-dimensional geometric topology.

Neil Gershenfeld

MIT Media Lab, Physics and Media Group, 20 Ames St., Cambridge, MA 02139, USA

E-mail:neilg@media.mit.edu

http://www.media.mit.edu/ neilg/

Prof. Neil Gershenfeld leads the Physics and Media Group at the MIT Media Lab, and codirects the Things That Think industrial research consortium. His laboratory investigates the interface between the content of information and its physical representation, from building molecular quantum computers to building musical instruments for collaborations with artists ranging from Yo-Yo Ma to Penn & Teller. He has a BA in Physics from Swarthmore College, worked at Bell Labs using lasers for atomic and nuclear physics research, received a Ph.D. from Cornell University studying order-disorder transitions in condensed matter systems, and he was a Junior Fellow of the Harvard Society of Fellows.