Modeling the influence of performance controls on Violin Timbre

Alfonso Pérez Carrillo, Jordi Bonada

Music Technology Group, Universitat Pompeu Fabra
alfonso.perez@upf.edu

Abstract. By means of a sensing system we are able to capture bowing and fingering actions executed by a violinist during real performances. The aim of this research is to model the relation between those actions and the sound produced. We describe the process for training and optimizing the model by means of neural networks. Given a set of control actions, the model is able to predict the spectral envelopes of the harmonic and noisy components of the sound which are used for sound synthesis. The model is validated by comparing a real recording with the corresponding synthetic sound produced.

1 Introduction

Provided with a measuring system that captures gestural and audio data of real performances, a set of spectral features and bowing descriptors are computed. Neural networks are used to find a regression function among spectral and bowing descriptors. A generative model is built which is able to predict the timbre, given the set of controls at that frame. Timbre is represented as harmonic and residual spectral envelopes. Prediction of spectral parameters is also used in the commercial synthesizer *Synful*[4], but it follows indications in a musical score instead of bowing controls. The main application of the model is for sound synthesis. It can be used as a pure spectral model by filling the predicted envelopes with harmonic and noisy components (in similar way to [7]), or integrated within a concatenative synthesizer to make gesture based spectral transformations. In this case, predicted envelopes are applied to the selected samples as a time-varying filter. Synthesis is used as well for validating the model.

2 Data descriptors

A set of musical scores is designed which cover the most common violin playing contexts. Scores are performed with a sensing system able to capture bowing gestures synchronized with sound. Bow motion is obtained by means of a commercial 3D motion tracking system[5] and bow hair-ribbon deformation is measured with an own-built device based on strain gages[3]. After alignment and segmentation of the recordings we obtained a dataset of around 900.000 analyzed temporal frames. Temporal resolution is given by the sensing system sampling rate (240Hz).

2.1 Inputs: Performance controls

Model's inputs include the main bowing controls that affect timbre, as reported in the literature ([2], [1], [6]) and some additional ones, namely pitch, tilt and derivatives of the parameters. Among other parameters we obtain:

- **Bow transversal position (*bpos*).** It is the distance from the frog of the bow to the point in contact with the string. The range of values goes from zero at the frog to around 65 cm at the tip.
- **Bow velocity (*bvel*).** It is computed as the smoothed derivative of bow position. A second velocity descriptor is *absolute velocity* which is preferred for learning the models because is independent of the bowing direction.
- **Bow acceleration.** Very important at note attacks with bow direction change. Three similar descriptors are being compared: 1) *acceleration*, the smoothed derivative of bow velocity: $bacc = \frac{dbvel}{dt}$, 2) in absolute values: $|bacc|$ and 3) as the derivative of absolute velocity: $bacc2 = \frac{d|bvel|}{dt}$
- **Bow-bridge distance (*bbd*).**
- **Bow force ($bforce$).** It is the force exerted by the bow on the string. It is also very present at note attacks.
- **Bow force derivatives.** We calculate first and second derivatives of force in order to have information about force variation within a frame.
- **Pitch.** It is the fundamental frequency extracted from the audio. Another similar descriptor is finger-bridge distance ($fingerPos$), which indicates the position of the pressing finger in the string.
- **Tilt**. It is the angle between the plane defined by the bow hairs and the string, being angle zero when the hairs are all on the string. It is very much related to dynamics.

2.2 Output: Timbre

Audio is analyzed in the spectral domain. Timbre is modeled separately as harmonic and residual components. Both parts are represented as the energy in different frequency bands. We defined 40 overlapped bands with centers following a logarithmic scale inspired in perceptual models such as Mel scale. The energy of each band is estimated as the average of the corresponding frequency bins weighted by a triangular function.

3 Building the Model

We make use of Neural Networks (NN) to find a regression function that predicts the energy for all the 40 frequency bands given a set of performance controls. The performance of a network depends on the selection of the inputs, the configuration of the network (feed-forward, cascaded networks, recurrent networks, etc.) and its parameters (learning rate, etc.). Next we describe different networks starting from the more basic network configuration until the setup that gives the best results. Setups are evaluated using a 10-fold-cross validation and compared using the correlation coefficient.

Basic setup. The most straightforward model is a feed-forward network trained by back-propagation with a hidden layer, with the main performance controls as inputs which according to the literature [1] are *bbd*, *bvel* and *bforce*. As timbre is represented as a set of 40 energy values, we need several models (models$_{i}$_bj, $s = string$, $b = band$, $i = 1..4$, $j = 1..40$), one for each band(j) and string(i). Average correlation coefficients for this simple models after a 10 fold cross-validation is 0.75. Default parameters for training the networks are: *learning rate* = 0.3, *momentum* = 0.2, *epochs* = 150, *hidden layers* = 1 and *neurons in hidden layer* = (♯inputs + ♯outputs)/2.

Input selection. In Table 1 and Table 2 we show the prediction results considering different combinations of input descriptors. Notice how a priori equivalent descriptors such as *bvel* and |*bvel*| are performing differently. The set of inputs that better predicts the harmonic timbre is: absolute bow velocity, acceleration, bow force, derivative of bow force, second derivative of bow force, finger position and tilt.

Making bands interrelated: RMS relative models. Models until this point are predicting energy in each band independently of the others, but bands are actually very correlated. In a frame with low energy, all bands will have low energy. A high prediction error in a specific band can cause discontinuities in the spectral envelope and error may vary a lot depending on the band as models are independent. To avoid this, we make prediction relative to Root Mean Squared Energy (RMS). Prediction has to be made in two steps: 1) prediction of the RMS Energy at each frame, and 2) use the predicted RMS Energy as extra input of the models. The introduction of RMS energy as input improves a lot the prediction, and errors stay more constant for different bands and strings.

Adding temporal information Temporal information is very important specially for attacks and note transitions. The model is fed with temporal information by including derivatives of position and force, that is, information about how are those descriptors evolving at a specific frame. Another tested solution is to include as inputs for the models not only the descriptors of the actual frame, but also of previous frames, but performance of the models is quite similar.

Optimizing Network parameters Default parameters for training the networks are in Table **??**. Different values may help the network converge faster or even find better solutions. In Figure 1 we show the evolution of the correlation coefficient different learning rate (LR) values. Both models perform better with low values. The same test was done for different values of the parameter *moment* in the range $[0.05 - 4]$ with no big changes in the performance. For models with $LR = 0.1$ an increase in the number of neurons improves performance, whereas for models with $LR = 0.3$ the best values are obtained with only 5 neurons.

	basic		plus acceleration						plus pitch			
model ID	100	101	102	102.1	103	104	105	105.1	106	106.1	106.2	106.3
string	x	x	x	x	x	x	x	x	x	x	x	x
bbd	x	x	x	x	x	x	x	x	x	x	x	
beta												x
velocity	x		x		x		x		x			
abs(velocity)		x		x		x		x		x	x	x
acceleration			x	x					x	x	x	x
d(abs(vel))/dt					x	x						
abs(acceleration)							x	x				
forceN	x	x	x	x	x	x	x	x	x	x	x	x
dforceN												
ddforceN												
pitch									x	x		
fingerPos											x	
Harmonic correlation	0.7502	0.792	0.756	0.795	0.754	0.794	0.743	0.793	0.785	0.803	0.804	0.770
Residual correlation	0.41	0.384	0.37	0.338	0.371	0.319	0.375	0.358	0.297	0.333	0.333	0.349

Table 1. Input for different setups and its Correlation Coefficients (I)

	plus tilt				plus dforce				plus ddforce		plus rms		
model ID	107	107.1	107.2	107.3	108	108.1	108.2	108.3	109	109.1	110	110.1	110.3
string	x	x	x	x	x	x	x	x	x	x	x	x	x
bbd	x	x	x		x	x	x		x	x	x	x	
beta				x				x					x
velocity	x				x				x		x		
abs(velocity)		x	x	x		x	x	x		x		x	x
acceleration	x	x	x	x	x	x	x	x	x	x	x	x	x
d(abs(vel))/dt													
abs(acceleration)													
forceN	x	x	x	x	x	x	x	x	x	x	x	x	x
dforceN					x	x	x	x	x	x	x	x	x
ddforceN									x	x	x	x	x
pitch	x	x			x	x			x	x	x	x	
fingerPos			x				x						
tilt	x	x	x	x	x	x	x	x	x	x	x	x	x
rms											x	x	x
Harmonic correlation	0.786	0.804	0.809	0.778	0.801	0.809	0.808	0.781	0.803	0.813	0.82	0.83	0.713
Residual correlation	0.332	0.342	0.346	0.345	0.305	0.351	0.346	0.328	0.318	0.319	0.637	0.631	0.656

Table 2. Input for different setups and its Correlation Coefficients (II)

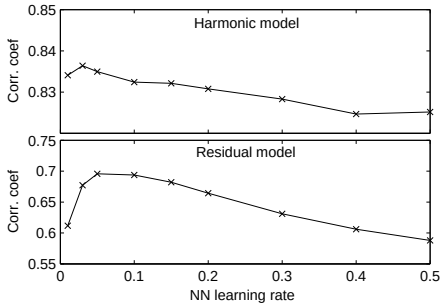

Fig. 1. Correlation coefficient for different values of the *learning rate*

Summarizing, the best network parameter values for the best harmonic model (model_ID=110.1) are: learning rate=0.05, moment=0.1 and number of neurons in hidden layer=15 with a correlation coefficient=0.85. In the case of residual models (model_ID=110.3) the values are: learning rate=0.03, moment=0.1 and number of neurons in hidden layer=17 correlation coefficient=0.7.

3.1 Other machine learning techniques

GMM and K-means. The mixture distribution parameters are optimized to obtain the maximum likelihood by means of the iterative expectation-maximization algorithm which stops when the likelihood increment during a iteration is lower than a threshold. We help the expectation-maximization algorithm perform faster, by applying previously a K-means algorithm to have a first estimate for the means and covariance matrices. Inputs are the same as for the Neural Networks. Default parameters for the algorithm are: *E-M iterations* = 3, *K-means iterations* = 100, *reg-eps*=$1e-4$, *do. diag*=0, *cov. factor*=1. Error values are almost the same independently of the training data size. But the number of gaussians seems to be very important, achieving the minimum error with 300 gaussians. More gaussians were not tested because of intensive memory usage, but it seems that we are close to an inflexion point in the error curve which would determine the optimal number. This quantity gives an estimation of the number of different timbre contexts found in the database and the mean and covariance matrix of each gaussian represents the distribution in the combined input+output space for a specific context. Gaussian are more informative than the neural networks and it would be interesting to explore those contexts. The correlation coefficient is about 0.81, similar to the neural networks.

Support Vector Machines. The parameters that define the model are: *Kernel Function* = *Radial Basis*, *Error*$< 10^{-4}$, C=100. The evaluation of the models shows a correlation coefficient of 0.81 which is quite similar to the other approaches.

4 Validation through Synthesis

Given a continuous sequence of control data the timbre model predicts the spectral envelopes of the violin sound at each temporal frame. Harmonic envelopes are filled with harmonic content corresponding to the pitch at that frame (pitch is taken from the score being synthesized) and residual envelopes are filled with noise. The result is a sinusoidal synthesis that follows a performance, its dynamics, timing, timbre and most of its nuances. This is a very simple synthesis but shows the potential of the timbre model. The model was also integrated in a concatenative synthesizer to improve the quality of the synthesis by allowing *gesture based timbre transformations*, that is, transforming the timbre of a sequence of samples, given its control's trajectories. Some sound samples can be found in a dedicated web page[1].

5 Conclusions

In this paper it was presented the procedure followed to build a violin timbre model based on neural networks, that is able to predict harmonic and residual spectral envelopes corresponding to a set of input control parameters. The model is used and validated by generating synthetic sounds that follow gestures of real performances. This allows to raise the degree of controllability and expressive capabilities in a sampled based synthesizer to a level close to that of physical models. A perceptual evaluation of the models is necessary to complement the numerical one presented here, and a refinement of the prediction during note attacks as well.

References

1. Anders Askenfelt. Measurement of bow motion and bow force in violin playing. *Journal of the Acoustical Society of America*, 80, 1986.
2. Lothar Cremer. *Physics of the Violin*. The MIT Press, November 1984.
3. E. Guaus, J. Bonada, E. Maestre, A. Pérez, and M. Blaauw. Calibration method to compensate bow tension variations in the prediction of bow force for real violin recordings. In *Proc. of ICMC*, submitted January 2009.
4. Eric Lindemann. Musical synthesizer capable of expresive phrasing. *Acoustical Society of America Journal*, 117:2700–2730, 2005.
5. Esteban Maestre, Jordi Bonada, Merlijn Blaauw, Alfonso Perez, and Enric Guaus. Acquisition of violin instrumental gestures using a commercial emf device. *International Conference on Computer Music*, 2007.
6. J.C. Schelleng. The bowed string and the player. *Journal of the Acoustical Society of America*, 1:53, 1973.
7. Bernd Schoner. *Probabilistic Characterization and Synthesis of Complex Driven Systems*. PhD thesis, MIT Media Lab, 2000.

[1] http://iua.upf.edu/∼aperez/MML