A GENERALIZED MUSICAL-TONE GENERATOR WITH APPLICATION TO SOUND COMPRESSION AND SYNTHESIS

Carlo Drioli *Davide Rocchesso*

Centro di Sonologia Computazionale - Dipartimento di Elettronica e Informatica
Università degli Studi di Padova, via Gradenigo, 6/A - 35131 Padova, Italy
driolie@mbox.vol.it roc@csc.unipd.it

ABSTRACT

A musical-tone generator based on physical modeling of the sound production mechanisms is presented. To the purpose of making this scheme general for a wide class of musical instruments, the nonlinear part of the tone-generator is modeled by a neural network. The system learns its parameters and the nonlinearity shape by means of nonlinear identification procedures based on waveform or spectral matching. Two possible applications of this model are discussed: sound compression can be obtained when considering the system as a nonlinear predictor, while sound synthesis can be obtained by adding control inputs to the network and by training the system to respond as desired.

1. INTRODUCTION

Synthesis by physical modeling has been broadly explored in the last years and many models reproducing different instruments have been proposed. Nevertheless, it is desirable to have schemes which are general enough for representing large families of instruments. This is a prerequisite for achieving efficient model-based compression and synthesis. We consider here a model which is sufficiently general to reproduce a wide class of musical instruments, namely the class of bowed-string and wind instruments, which can generate sustained tones and are characterized by a one-dimensional resonator. Examples of these instruments are found in the traditional orchestra strings and winds, such as the violin or the clarinet. These instruments are in principle more difficult to model than percussive instruments, since nonlinearities can never be neglected. Moreover, there is a non-negligible feedback mechanism between the nonlinear and the linear part.

Prior work on system identification of specific instruments has been developed in [1]-[5]. In [6] a procedure is given for identifying the parameters of a Karplus-Strong-like model, which is suitable for percussive sounds reproduction.

In this paper we present an identification procedure of the nonlinear part based on the Genetic Algorithm (GA) [7] applied to the parameters of a Radial Basis Function (RBF) network [8], which acts as a generalized nonlinear map.

2. THE GENERALIZED MUSICAL-TONE GENERATOR MODEL

In all of the instruments under consideration, there is a linear part, the resonator, which interacts with a nonlinear element, called the exciter. The resonator models the part where vibrations propagate, the exciter is the part responsible for creating and sustaining the oscillation [9]. Our reference model (Figure 1) belongs to the class of waveguide synthesis models [10], since the resonator is represented with a delay line and some filtering elements. The delay line determines the fundamental periodicity of sound, while the filters take care of effects of losses (Low-Pass filter H_{LP}), tuning adjustment and dispersion (All-Pass filter H_{AP}). This kind of one-dimensional resonator is applicable to wind and bowed-string instruments [11]. The novelty of our model resides in the nonlinear element, the exciter, which is intended to be as general as possible. In classical synthesis by physical models, the exciter is represented as a nonlinear instantaneous map with, possibly, a dynamic, linear part. The map is very dependent on the kind of excitation we are considering, and in the musical acoustics literature one can find various maps for reeds, jets, bows, etc. We decided to adopt an instrument-independent map, and to realize it by means of a RBF network, that is a one-hidden layer network capable of approximating any continuous function if a sufficiently high number of hidden units is used.

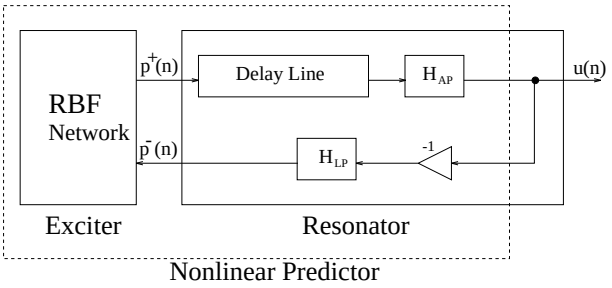

Figure 1. The Exciter-Resonator interconnected scheme used to model a generalized musical-tone generator. The dashed box points out the nonlinear predictor that will be used for sound compression.

Once the model is given, a procedure for identifying the model's parameters is needed. The nonlinear optimization procedure that we adopted is the GA, where each chromosome of the population is encoded by a string of real numbers, say the RBF network's parameters (centroids, shapes and heights of the RBF's [8]) together with the coefficients of the filtering elements of the resonator [12]. The *fitness* of the j-th chromosome is then evaluated in terms of the estimation error e_j, defined as

$$e_j = \sum_{n=1}^{N} (u_d(n) - u_j(n))^2 \qquad (1)$$

where N is the window size over which the error will be

accumulated, $u_d(n)$ is the desired output and $u_j(n)$ is the reconstructed output of the j-th system. In order to deal with the real-valued chromosome, a set of proper operators has been used [13]:

- *Selection*: among all the individuals of the population, the ones with lower estimation error (higher fitness) are selected to survive and to be randomly paired off for new chromosome generation. The percentage of surviving chromosomes is usually fixed at 15-20% of the entire population.

- *Crossover*: let $\mathbf{w}$ and $\mathbf{v}$ the parent arrays selected for mating. The new chromosome $\mathbf{z}$ generated by the two parents can be written as $\mathbf{z} = \mathbf{w}(a-1) + \mathbf{v}a$, with $a \in [0,1]$. It is thus a linear interpolation of all of the two arrays entries. The operator can also be modified to randomly select sections of the arrays and perform a partial linear interpolation.

- *Mutation*: to prevent the convergence to local minima (non optimal solutions) this operator can randomly change the value of each chromosome's entry, although with very low probability. If v_{ij} is the j-th entry of the i-th chromosome and it has been selected for mutation, the result of the operation is a new entry $\tilde{v}_{ij} = v_{ij} + d$, where d is a random value ranging from $V_m - v_{ij}$ to $V_M - v_{ij}$ (if $[V_m, V_M]$ is the admitted range for that entry).

- *Extinction and Immigration*: this operator acts when the estimation error tends to stagnate due to the fact that, after several generations, the chromosome pool can became homogeneous and mutation is not efficient anymore. Extinction eliminates all of the chromosomes in the current generation but the one corresponding to the minimum estimation error. These individuals are then replaced by a set of randomly generated ones.

As a first example, we consider the identification of model parameters starting from a sound signal generated by a model as simple as that of Figure 1, but having a nonlinear map which is stored in a look-up table. The target nonlinearity adopted has been usefully used for simulating the clarinet [11]. This is a sort of minimal-requirement test, for assessing the correctness of the procedure. We were able to recreate the expected tone using as few as six RBF's, while the resonator was kept the same as in the target model (and no All-Pass filter was considered). The original model was driven by a step in the mouth-pressure control signal, so that the nonlinearity remains fixed along the whole sound length. The results are reported in Figure 2. It is interesting to notice that the tone can now be represented by less than twenty model parameters. This procedure can be considered as a sound analysis and synthesis tool, since it can be used to evaluate the instrument excitation nonlinearity given a typical waveform. This information can then be used to resynthesize instrumental characteristics. We stress the fact that, as long as the resonator is not known in advance, the reconstructed nonlinearity may differ significantly from the original one, since some form of compensation between the linear and nonlinear part may occur in the minimization procedure.

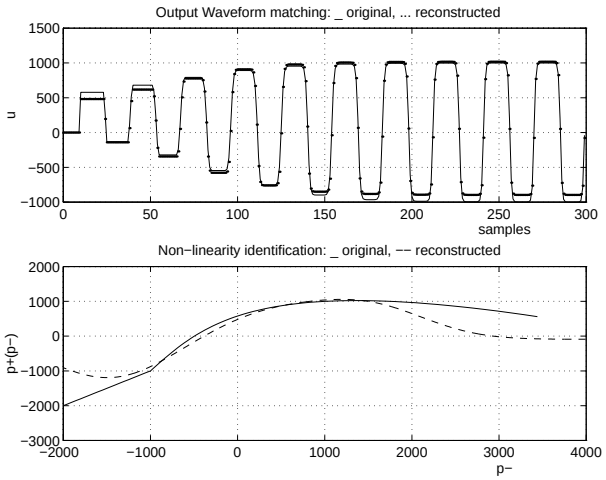

Figure 2. Identification of a known excitation nonlinearity. The RBF Network hidden layer contains six gaussian units. Note that nonlinearity identification is good within the range interested by the signal.

3. APPLICATION OF THE TONE-GENERATOR TO SOUND COMPRESSION

The main goal of sound compression is to obtain a compact digital representation of sound signals for the purpose of efficient transmission or storage. Techniques for sound compression can be broadly divided into two main categories: perception-based and production-based techniques. The first category is based on exploitation of the perceptual properties of the human auditory system (e.g. direct quantization or MPEG/audio) and requires no further information on the sound production mechanisms. On the other hand, perception-based techniques implies the existence of a suitable model able to well represent the phenomena underlying the production of the given sound. This kind of approach has been broadly used for speech coding, since the physics and geometry of the speech production mechanisms are well known, and there are models which represent them reliably [14].

3.1. Compression Algorithm

The use of a physical model for sound compression purposes leads to a Predictive Quantization scheme [15], as illustrated in Figure 3. The Prediction block in our case is a nonlinear predictor, realized by means of the entire interconnected system presented in the previous section (dashed box in Figure 1), while quantization is conducted on the residual error $e(n) = u(n) - u_d(n)$. From the scheme of the nonlinear predictor in the dashed box it is also clear that the closed loop computability is respected, due to the m-samples delay line.

In the compression process that we explored, the parameters of the model (RBF network weights and centroids, and H_{AP} filter coefficients) are extracted by explicitly minimizing a measure of the difference between the original signal $u_d(n)$ and the predicted signal $u(n)$. It is therefore an *analysis-by-synthesis* process. The quantized residual error $\hat{e}(n)$ evaluated at compression time can be considered, together with the predictor parameters, as the compressed version of the signal: in fact the output of the tone generator can be filled with $\hat{e}$(n) in order to obtain the correct

signal at reproduction time, with a reproduction error given by $e_r(n)=\hat{u}_d(n)-u_d(n)$. Note that even if this approach does not directly minimize the difference between desired signal $u_d(n)$ and reconstructed signal $\hat{u}_d(n)$, the reconstruction error is controlled since it is easy to prove that

$$E[(e(n)-\hat{e}(n))^2] = E[(u_d(n)-\hat{u}_d(n))^2] \qquad (2)$$

The minimization procedure that we adopted is the GA, where the chromosome is encoded as in the previous identification example with the addition of the allpass filter coefficients. We stress the fact that the compression is conducted by frame segmentation of the input signal. A frame analysis scheme takes into account slowly time-varying phenomena, and allows the evaluation of system parameters at a frame rate lower than the sample rate. Moreover, the spectral content of the residual error can be exploited when analysing periodic signal segments: it is possible to update the error signal and the parameters (evaluated at a starting frame i) after n_f frames, where the (n_f+1)-th frame is the first one presenting an unacceptable output deviation (n_f will be variable and determined run-time). This technique allows to save computation time when parameter evaluation is not needed at each frame, and raises compression rate too.

A further quality improvement can be obtained evaluating, for the frames following the frame i, just a *differential* residual error signal $e'_i(n)$, that can be added to the loop at reproduction time. The total residual error for the generic k-th frame following the frame i is then

$$e_{i,k}(n) = e_{i,k-1}(n) + e'_{i,k}(n) = \sum_{j=0}^{k} e'_{i,j}(n), \qquad (3)$$

where $e'_{i,0}(n) = e_i(n)$.

3.2. Compression Results

In this section we present some results obtained with the given compression algorithm. In all cases the neural network used is a four-hidden-units RBF network and H_{ap} is a fourth order All-Pass filter. In Figure 4 and Figure 5 two attack frames of respectively a clarinet and a violin tone are shown. It is interesting to note how the excitation nonlinearities of the two instruments differ in the range interested by the signal (as we expected, the violin tone requires a much more severe deviation from linearity). In Figure 6 a comparison between the original and residual signal is given for a pitch-varying clarinet tone. Figure 7 shows how the nonlinearity acts to compensate for the lack of a more complete physical model of the clarinet keys when analyzing a pitch transient. In this case frame length and delay line length evaluation was made by hand, and the two considerable spikes visibles in the residual error are due to changes in the delay line in correspondence of the beginning of two transient frames.

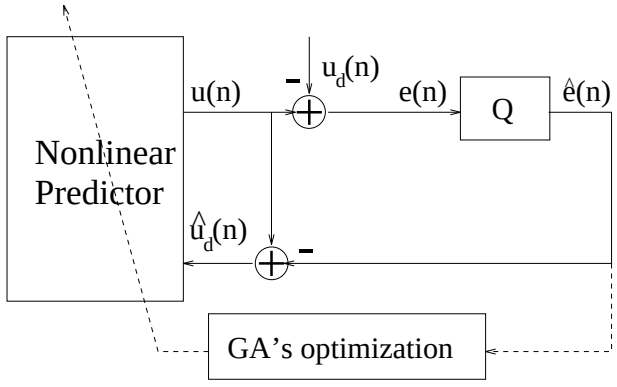

Figure 3. Nonlinear Predictive Quantizer at compression-time.

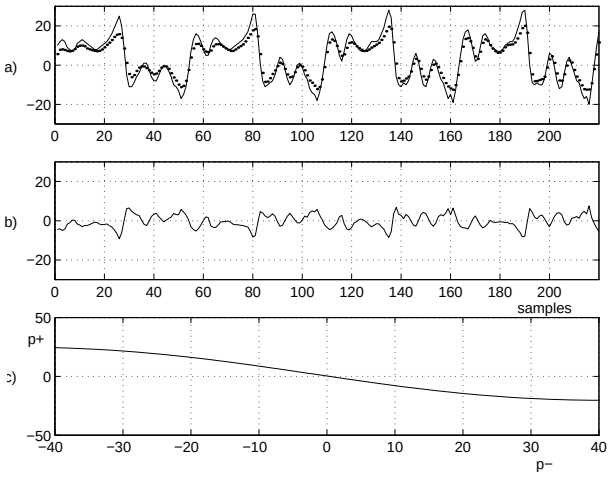

Figure 4. Attack frame of a clarinet tone: a) the original output u_d (solid line) vs. the predicted output u (dotted line); b) residual error; c) exciter nonlinearity.

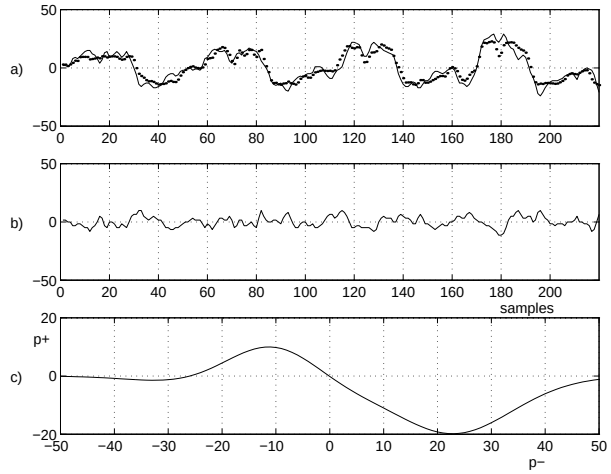

Figure 5. Attack frame of a violin tone: a), b) and c) same as in Figure 4.

4. APPLICATION OF THE TONE-GENERATOR TO SOUND SYNTHESIS

The system identification approach illustrated for sound compression purposes can be adapted to a sound synthesis context. However, there are some important differences that are worth emphasizing. First of all, a synthesis device requires one or more control inputs to drive the excitation block. This leads to a RBF network with more than just one

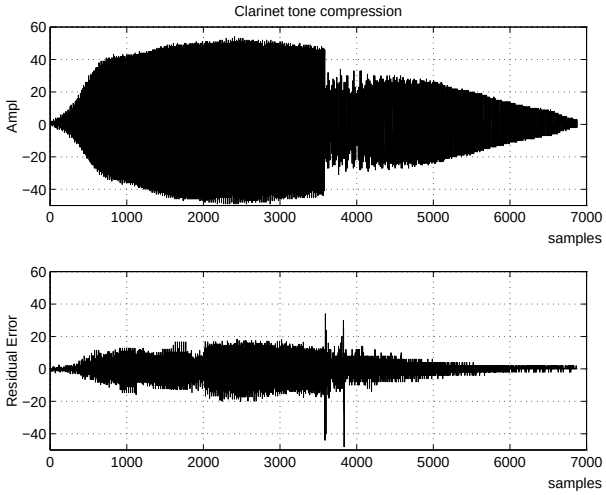

Figure 6. Compression of a pitch-varying clarinet tone

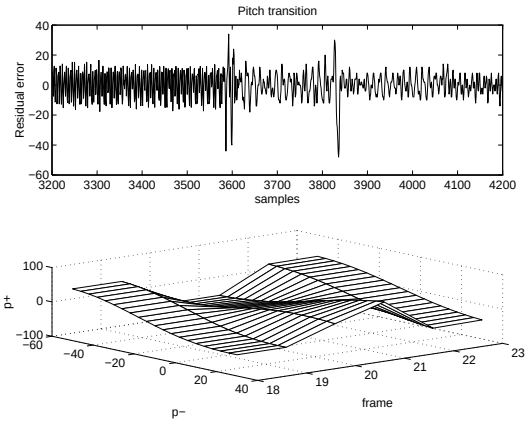

Figure 7. Detail of the pitch variation segment

input node, reproducing thus a multidimensional hypersurface. In principle, the whole hypersurface can be modeled by training the system with a trining set of cause-effect examples covering all the instrumental most peculiar characteristics. The second important difference resides in the more difficult integration of a correction mechanism based on the residual error added to the loop. In all the experiments we made, the latter seemed to be the most crucial point to deal with. Due to the intrinsic simplifications of the model that we adopted, we were able to reproduce just very simple waveforms. Further improvements might come from the insertion of dynamics in the excitation block. In a sound synthesis context, the possibility to maintain the residue-based correction mechanism shows up if we consider the residual error as an excitation signal. In this case, during the training of the system, the construction of a residual codebook must be performed. Thereafter, at reproduction time, the right excitation can be selected to reproduce the desired tone.

5. CONCLUSIONS

The model presented in this paper should be general enough to represent various musical instruments, in both contexts of sound compression and sound synthesis. A procedure for performing sound compression is proposed and some results of system identification are given. The base model, used as a nonlinear predictor, has been kept as simple as possible to be computationally efficient. However, further model improvements are expected in order to simulate with better accuracy the sonic behavior of actual instruments. Along this line, the main purposes for future research are: (1) having lower prediction errors while compressing a musical tone, and (2) having the ability of reproducing complex waveforms when synthesizing a musical instrument.

REFERENCES

[1] J. Vuori and V. Välimäki, "Parameter estimation of non-linear physical models by simulated evolution-application to the flute model," Proc. Int. Comp. Music Conf., pp. 402-404, Tokyo 1993.

[2] M. Karjalainen, V. Välimäki and Z.Janosy, "Towards high-quality synthesis of the guitar and string instruments," Proc. Int. Comp. Music Conf., pp. 56-63, Tokyo 1993.

[3] J. Laroche and J. M. Jot, "Analysis/Sinthesis of Quasi-Harmonic Sounds by use of the Karplus-Strong Algorithm," Journal de Phisique IV; C1, pp. 117-120, 1992.

[4] P. R. Cook, "Non-Linear Periodic Prediction for On-Line Identification of Oscillator Characteristics in Woodwind Instruments," Proc. Int. Comp. Music Conf., pp. 157-160, Montreal, Canada 1991.

[5] J. O. Smith, "Techniques for Digital Filter Design and System Identification with Application to the Violin," PhD Thesis, CCRMA Stanford University 1983, Report N. STAN-M-14.

[6] G. Evangelista and S. Cavaliere, "Karplus-Strong Parameter Estimation", Proc. XI Colloquio di Informatica Musicale, pp. 85-88, Bologna, Italy 1995.

[7] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," NY, Addison Wesley, 1989.

[8] T. Poggio and F. Girosi, "Networks for Approximation and Learning," Proc. IEEE vol. 79, n. 9, pp. 1481-1497, 1990.

[9] G. Borin, G. De Poli and A. Sarti, "Sound Synthesis by Dynamic Systems Interaction," in Readings in Computer-Generated Music, pp. 139-160, IEEE Comp. Soc. Press, D. Baggi ed., 1992.

[10] J. O. Smith, "Physical Modeling Using Digital Waveguides," Computer Music Journal, vol 16, n. 4, pp. 74-91, 1992.

[11] D. Rocchesso and F. Turra, "A Generalized Excitation for Real-Time Sound Synthesis by Physical Models," Proc. Stockholm Music Acoustic Conf., pp. 584-588, 1993.

[12] L. Yao and W. Sethares, "Nonlinear Parameter Estimation via the Genetic Algorithm," IEEE Trans. Signal Processing, vol. 42, n. 4, pp. 927-935, 1994.

[13] Z. Michalewicz, "Genetic Algorithms+Data Structures=Evolution Programs," Springer Verlag, 1994.

[14] A. S. Spanias, "Speech Coding: A Tutorial Review," Proc. IEEE, vol. 82, n. 10, pp. 1541-1582, 1994.

[15] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression," pp. 203-220, Kluwer Academic Publishers, Boston/Dordrecht/London, 1992.