Automatic Estimation of Control Parameters: An Instance-Based Learning Approach

Wim D'haes, Xavier Rodet {dhaes, rod}@ircam.fr
IRCAM – CENTRE GEORGE POMPIDOU
1, place Igor-Stravinsky · 75004 Paris · France

Abstract

An automatic method is proposed for the estimation of control parameters for musical synthesis algorithms. This method can be applied on a large class of systems for which a simulation, or "model" is available. In this paper we describe its application on a physical model of a trumpet where the "system" is an acoustical instrument, a trumpet, and the "model" a computer program that simulates trumpet sounds. An instance based learning program was developed that determines the control parameters of the physical model from a set of characteristics of the desired sound. The main advantage of this approach is that it can be applied to any synthesis algorithm.

1 Introduction and State of the Art

When using a physical model to simulate an acoustic instrument the way it is controlled is as important as the quality of the model itself. A physical model that is potentially capable of simulating any sound of an acoustic instrument will still sound very unnatural if it is not controlled correctly. Therefore, in order to produce realistic approximations of instrument tones, it is important to use appropriate time-varying control functions. A real-time implementation of the model controlled by an adapted instrument-like interface may provide a preliminary solution, however it is not known how the interface parameters must be mapped to the control parameters of the synthesis model. In addition, it is impossible for a musician to control such an interface in order to obtain a professional musical performance. Therefore, techniques that can determine the control parameters in order to simulate a given sound are very interesting.

Although one approach to finding control parameters would be to invert the mathematical equations on which the model is based (Helie et al. 1999), the method proposed in this article considers the model as a "black box" neglecting the inner workings of the system. Only the output of the system is observed for different inputs. The method is described specifically for the parameter estimation of musical synthesis algorithms, but can be applied on any system-model couple for which the inputs and outputs are recordable and small in number. If the parameters of the model can be calculated so as to reproduce accurately a recorded trumpet performance of a professional musician playing a high quality instrument, very interesting applications arise.

Related research involving the control of synthesis algorithms, waveform synthesis in particular, has been done by Dannenberg and Derenyi (1998). Neural networks and memory based machine learning were used in the work of Wessel et al. (1998). In these examples a *signal model*[1] is used, controlled by fundamental frequency and amplitude envelope functions. Our article addresses the problem of the control of *physical models* using continuous time-varying parameters that have a (more or less) physical meaning but are less directly related to characteristics of the produced sound. A similar technique is Code-Excited Linear Prediction (CELP) speech coding where the most appropriate innovation sequence is searched from a code book to optimize a given similarity criterion (Schroeder and Atal 1985).

2 Characterization and Metric

Consider a system S which is controlled by a set of parameters $p(t)$ and produces a signal $s(t) = S(p(t))$. This system is simulated by a model X with similar (but not necessarily exactly the same) control parameters $q(t)$ which produces a signal $x(t) = X(q(t))$. The main goal of the proposed method consists of determining the control parameters $q(t)$ for a given sound $s(t)$ such that the corresponding sound $x(t) = X(q(t))$ is most similar to the original signal $s(t)$ (see figure 1).

In order to determine how similar the original signal and its resynthesis are, a metric must be developed. This distance measure is defined in terms of perceptually relevant characteristics $K(s(t))$ that are estimated from the signal $s(t)$ for regular time intervals. In the case of the trumpet the sustained part of the sound is harmonic and can be described appropriately by its fundamental frequency and a characterization of its spectral envelope, such as linear prediction coefficients or cepstrum coefficients. Since the cepstrum coefficients do not adequately represent the spectrum of a sound with a high

[1] as opposed to a physical model

pitch, the *discrete cepstrum* is preferred, which is calculated from peaks in the short time spectrum (Galas and Rodet 1991, Schwarz and Rodet 1999). The curve defined by the discrete cepstrum is divided in eight equal bins according to the Bark scale yielding a feature vector of eight elements that indicate the similarity between two spectra.

Unfortunately, most methods that determine the fundamental frequency fail when transients (fast variation in amplitude or frequency) occur. In this case the fundamental frequency and the spectral envelope are unreliable, resulting in a false characterization of the signal. This problem will be addressed in more detail in section 5.

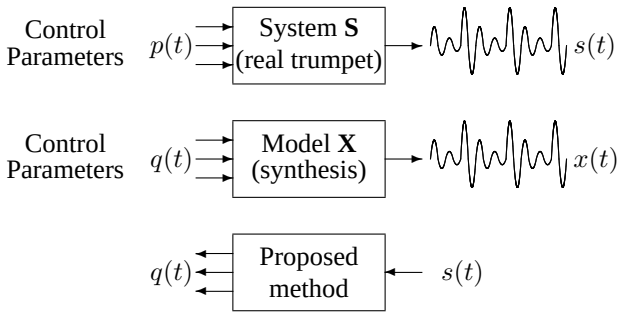

Figure 1: Schematic representation of the problem

3 Instance-Based Learning

After the calculation of the sound characteristics, the estimation of the control parameters can be seen as the modelling of a multi-dimensional function f that takes as input the characteristics of the signal $K(s(t))$ and returns the control parameters $q(t) = f(K(s(t)))$. The pattern recognition field offers powerful techniques to model this function from a *training set* of feature vectors containing input and output values (Fukunaga 1990, Bishop 1995). In contrast to *classification* problems where one seeks to approximate the probabilities of membership to a number of classes, this is a *regression* problem since the values of $q(t)$ are continuous. In general, three alternative approaches can be distinguished. For the *parametric* approach a specific functional form is assumed containing a number of parameters which are optimized in order to fit the data set. By contrast, the second technique of *non-parametric* estimation does not assume a functional form, but allows the form to be determined entirely by the data. The third approach, called *semi-parametric* estimation allows a very general class of functional forms, for instance neural networks, in which the number of adaptive parameters can be increased in a systematic way.

Since no mathematical form for f can be assumed, a non-parametric *k-nearest neighbors* estimation technique was used. This technique searches the most similar feature vector in the data set and returns the corresponding output. This technique is also known in the field of *machine learning* where it is called *instance-based learning* (Mitchel 1997). The learning procedure consists simply of storing new input-output pairs in the database.

A disadvantage of the instance based approach is that it requires a large number of distance computations in order to find the k-nearest neighbors. Therefore, a *branch and bound search algorithm* is used that facilitates the rapid calculation of the k nearest neighbors (Fukunaga and Nerada 1978, Niemann and Goppert 1988). A branch and bound algorithm is a tree search algorithm that uses a *hierarchical decomposition* of the sample set of vectors. It does not compute a distance to all patterns in the data set but only to a certain node in the tree representing a subset of the sample. By using proper rules it is decided whether a pattern in this node can be a nearest neighbor. If this is not the case, the complete node and all patterns belonging to it may be discarded.

4 Implementation

4.1 The Physical Model

At IRCAM a physical model of a trumpet was developed (Vergez 1999, Vergez and Rodet 2001) and implemented in the real-time environment jMAX (Tisserand 1999). A trumpet consists of a mouthpiece on which the player places his lips, followed by a tube of variable length (different finger positions) that finally ends into the bell. The lips are modeled by a mass-spring system to which a non-linearity is introduced when the lips close. The tube is simulated by a transfer function measured from a real trumpet and a delay that depends on the tube length. Finally, the effect of the bell is modeled by a high pass filter.

The most important control parameters $q(t)$ of the model are: the pressure in the mouth, the frequency of the lips, the viscosity of the lips and the length of the tube. The relationship between the tube length and the lip frequency is very important since it determines which mode of the tube is excited. The model can be controlled in real time by a sax MIDI (Yamaha WX7) which produces MIDI events that are then converted to the control parameters of the model. Another possibility is to produce an SDIF file that contains the control events. Each SDIF frame contains one matrix containing the type and value of the event. In doing so, every possible control parameter evolution can be generated and used to control the physical model.

4.2 Data Set Production

Two types of data sets were generated for the inversion of the physical model. By controlling the real-time implementation of the model, a first training set of sounds was generated recording the sound and control parameters. All notes on the chromatic scale were played with a slow crescendo and diminuendo in order to have all possible intensities. Then a slow vibrato was added so that more variation in timbre

could be achieved. By contrast, the second set was obtained by *sampling* the control parameter space. Slow crescendos were obtained by augmentation the mouth pressure for every combination of lip frequency, lip viscosity and tube length.

4.3 The Learning Phase

As stated before, the learning phase for instance-based learning consists simply of storing new input-output couples in a database. For the estimation of the control parameters the input parameters are the characteristics $K(x(t))$ of the sound, and the output the control parameters of the model $q(t)$. On the test set of sounds an additive analysis was applied extracting the necessary characteristics or features. These features are the fundamental frequency, and the spectrum characterization. Then, the branch and bound algorithm described in section 3 was developed, facilitating a rapid search of the nearest neighbors. The efficiency of the search algorithm is an important issue since the control parameters must be determined 100 times for one second of sound.

4.4 The Simulation Phase

The simulation of a recorded sound $s(t)$ is achieved by the following steps:

1. The characteristics $K(s(t))$ of the sound are calculated at a frame rate of 100 Hz.
2. For each frame the vector with the most similar characteristics $K(x(t))$ is searched in the database returning the corresponding control parameters $q(t)$.
3. These parameters are written to file and used for the resynthesis of a sound $x(t)$ that is close to the original sound $s(t)$.

5 Results and Discussion

5.1 Simulations

For the first data set that was produced with the real-time implementation of the physical model, different sounds were simulated starting with sounds that were part of the training set. For these sounds the simulation was very similar since the same exact feature vectors were available in the database. However, when attempting to simulate a sound that was not represented well in the data set several problems occurred due to sparse regions in the feature space. This sparsity can cause that sound characteristics that vary significantly around one isolated feature vector return always the same control parameters. In this case the resynthesized sound will remain stable while the original sound varies dynamically. By contrast, a small change of the signal characteristics may yield large variations in control parameters when the two closest feature vectors are very distant from each other.

Finally, recordings of an acoustic instrument were simulated. Since the training set contains only a chromatic scale of notes, and thus only a few discrete fundamental frequencies are well represented in the data set, a problem might occur when the recording of the real trumpet is not in tune with the scale that was played in the training set. In order to solve this problem a histogram for both the database and the sound file was made. The pitch feature of the original sound was adjusted slightly in order to guarantee that the feature vectors did not fall in a sparse region of the feature space. After some corrections of the control parameters at the transients a satisfactory simulation was obtained.

In order to avoid the sparse locations in the data set, a second set was generated by a uniform sampling of the control parameter space. For the shortest tube length (no valves pressed) crescendos were generated for lip frequencies that excite the sixth mode of the tube. This was repeated for a number of values for the lip viscosity. A trumpet sound with a close fundamental frequency was simulated that contained dynamic variation in amplitude and a considerable vibrato. During the first simulation, it was observed that the variation in the fundamental frequency was obtained by changing the length of the tube. However, this would be impossible on a real instrument. Therefore, the length of the tube was determined first for each note. Then, when the other control parameters were determined, the only part of the data set considered was that produced by this fixed tube length.

5.2 Transients

The approach that is described above assumes implicitly that the relationship between the control parameters and the sound characteristics is instantaneous or time-independent. This assumption is valid when the control parameters are varied slowly, but when the parameters are changed rapidly it takes several frames for the model to converge to a stable sound. As a result, very similar characteristics could yield very different control parameters which would be simply unacceptable. One solution might consist of considering consecutive frames as a single feature vector permitting to capture the dynamic evolution of the control parameters. This will result in enormous data sets since the quantity of training data needed to specify the mapping grows exponentially with the dimensionality of the input space.

A second problem is that when transients occur, the characteristics that are estimated from the sound are very unreliable resulting in a false characterization of the signal. In this case, the control parameters cannot be retrieved by the proposed inversion method. However, experiments with the real-time implementation prove that the model is able to produce very natural transients. These transients can be detected by observing the fundamental frequency and the energy of the signal. Afterwards, the control parameters during the transients can then be extrapolated from its context. In the case

of the trumpet two main types can be distinguished: a sudden augmentation or diminution of the pressure (onset or offset), and a sudden change in tube length and lip frequency (slur). It is assumed that during the onset the tube length and lip frequency used is the same as during the stable part of the note. During a slur, the lip frequency and tube length are changed instantaneously while the pressure is interpolated linearly. This results in very natural sounding transients.

6 Conclusions and Further Work

Although the databases that are used for the inversion are limited, very promising results were obtained using the control parameter estimation technique described in this paper. This technique can only be applied for the stable parts of the sound, but the control parameters during the transients can also be determined automatically from its context using a very simple extrapolation. The physical model produces very realistic transients when the control parameters are changed instantaneously. Furthermore, one must keep in mind that the acoustics of the environment are not taken into account, although this is perceptually relevant. The physical model calculates only the signal at the bell of the instrument. Ideally, special recordings can be used that are made in an anechoic chamber with a microphone at the bell of the trumpet.

Since the performance of the inversion relies completely on the data set (data-driven) future work will consist of designing much larger data sets. In doing so, coverage of the entire sound space that can be produced by the model will be more complete. When a more uniformly sampled data set is produced, interpolation between several close vectors can be applied resulting in finer variations of the control parameters. Another interesting future research direction consists of adding *iterative optimization* to the current system. In this case the control parameters are changed iteratively in a way that optimizes the similarity criterium.

7 Applications

There are accurate parameter estimation techniques for several synthesis algorithms that are based on a signal model. With these algorithms, a recorded sound can be analyzed and resynthesized with modified control parameters resulting in sound manipulations of a very high quality. This paper attempts to answer the need for control parameter estimation methods for physical models. When a satisfactory synthesis of a recorded instrument can be obtained from a physical model, a composer can manipulate the sound by modifying the gestures of the musician. In addition, one also has the possibility to change the sound characteristics and determine the control parameters that realize this modification.

Finally, another very promising application would be compression. The control parameters can be sampled using a very low sampling frequency, say 100 Hz, or by asynchronous events. In the case of the trumpet we have four independent control parameters coded each by a 16 bit integer yielding 800 bytes/sec. The quality however will not be effected since it is guaranteed by the synthesis algorithm. Furthermore, the control parameters are often slowly varying and can thus be coded again very efficiently, achieving even higher compression rates.

8 Acknowledgements

This work was financially supported by IRCAM, the University of Antwerp (Visionlab) and the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT), Brussels.

References

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Dannenberg, R. B. and I. Derenyi (1998, September). Combining instrument and performance models for high-quality music synthesis. *Journal of New Music Research*, 211–238.

Fukunaga, K. (1990). *Statistical Pattern Recognition*. Academic Press.

Fukunaga, K. and P. M. Nerada (1978, July). A branch and bound algorithm for computing k-nearest neighbors. *IEEE transactions on computers*, 750–753.

Galas, T. and X. Rodet (1991). Generalized functional approximation for source filter system modeling. *Proceedings of Eurospeech*, 1985–1088.

Helie, T., C. Vergez, J. Levine, and X. Rodet (1999). Inversion of a physical model of a trumpet. *ICMC*, 149–152.

Mitchel, T. M. (1997). *Machine Learning*. McGraw-Hill International Editions.

Niemann, H. and R. Goppert (1988, February). An efficient branch and bound nearest neighbour classifier. *Pattern Recognition Letters*, 67–72.

Schroeder, M. R. and B. S. Atal (1985). Code-excited linear prediction (CELP): High quality speech at very low bit rates. *ICASSP*, 937–940.

Schwarz, D. and X. Rodet (1999). Spectral envelope estimation and representation for sound analysis-synthesis. *ICMC*, 351–354.

Tisserand, P. (1999). Portage du modèle physique de trompette sous jMAX. Technical report, IRCAM.

Vergez, C. (1999). *Trompette et trompettiste: un système dynamique non linéaire à analyser, modéliser et simuler dans un contexte musical*. Ph. D. thesis, Univ. Paris 6, IRCAM.

Vergez, C. and X. Rodet (2001). Trumpet and trumpet player: Modeling and simulation in a musical context, (to appear). *ICMC*.

Wessel, D., C. Drame, and M. Wright (1998). Removing the time axis from spectral model analysis-based additive synthesis: Neural networks versus memory-based machine learning. *ICMC*, 62–65.