CENTER FOR COMPUTER RESEARCH IN MUSIC AND ACOUSTICS
AUGUST 1997

DEPARTMENT OF MUSIC
REPORT NO. STAN-M-101

CCRMA PAPERS PRESENTED AT THE 1997 INTERNATIONAL COMPUTER MUSIC CONFERENCE THESSALONIKI, GREECE

Jonathan Berger, Chris Chafe, Alex Igoudin, David Jaffe, Tobias Kunze, Scott Levine, Fernando Lopez-Lezcano, Sile O'Modhrain, Pat Scandalis, Gary Scavone, Julius Smith, Tim Stilson, Heinrich Taube, Scott Van Duyne, Tony Verma

CCRMA
DEPARTMENT OF MUSIC
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-8180

TABLE OF CONTENTS

A Neural Network Model of Metric Perception and Cognition in the Audition of Functional Tonal Music.

Jonathan Berger
CCRMA
Stanford University
Stanford, CA 94305, U.S.A
brg@ccrma.stanford.edu

Dan Gang *
Institute of Computer Science
Hebrew University
Jerusalem 91904, Israel
dang@cs.huji.ac.il

Abstract

In our previous work we proposed a theory of cognition of tonal music based on control of expectations and created a model to test the theory using a hierarchical sequential neural network. The net learns metered and rhythmecized functional tonal harmonic progressions allowing us to measure fluctuations in the degree of realized expectation (DRE). Preliminary results demonstrated the necessity of including metric information in the model in order to obtain more realistic results for the model of the DRE. This was achieved by adding two units representing periodic index of meter to the input layer. In this paper we describe significant extensions to the architecture. Specifically, our goal was to represent more general meter tracking strategies and consider their implications as cognitive models. The output layer of the sub-net for metric information is fully connected to the hidden layer of sequential net. This output layer includes pools of three and four units representing duple and triple metric indices. Thus the sub-net was able to influence the resulting DRE, that was expected by the net. Moreover, by including multiple metric parsings in the output layer the net reflects conflicts between parallel possible interpretations of meter. This output was fed back into the sub-net to influence the next predictions of the DRE and the meter. In addition, the target harmony element was fed into the context instead of the actual output, thus simulating the interactive influences of harmonic rhythm and meter.

1 Introduction

"The poets have a proverb: Metra parant animos (the emotions are animated through verse). They say such quite rightly: for nothing penetrates the heart as much as a well-arranged rhyme scheme [Mat39]".

Johann Mattheson's awareness of the cognitive power of underlying metric temporal patterns (be it musical metric feet or rhythmic modes) in music and poetry has been consistently stated and, over the past century, empirically researched. That listening to music involves an initial creation of a metric schema has been well documented. What is not clear, however, is the process in which the listener arrives at a working schema.

In this paper we explore and model a possible scenario of metric decision making. As a point of departure we incorporate observations, speculation, and perceptual studies that suggest:

1. Constructing a metric schema is a task critical to music audition. In Mattheson's words "...the ordering of the feet in poetry and the well-constructed alternation of meters, even if there were no rhyme scheme, produces something initially so certain and clear in the hearing that the mind enjoys a secret pleasure from the orderliness and accepts the performance so much the easier."

2. Listeners of Western music have preconceived organizational schemas grouping into duple or triple metric units. Listeners count in hierarchies (base 3 or base 4 for most common meters). [Pov81] demonstrated that untrained listeners can accurately distinguish between duple and triple metric units. Furthermore, considerable evidence of preconceived grouping preferences suggest that this is applicable to meter recognition.

*Dan Gang is supported by an Eshkol Fellowship of the Israel Ministry of Science

Although generative algorithms (e.g., [LHL82]) and autocorrelative methods (e.g., [DH89]) for meter recognition are successful in their task they do not offer a plausible explanation of how a listener applies schematic based expectation of duple or triple groupings to determine meter. The music theory literature regarding meter (e.g., [LJ83]) similarly fail to account for this basic task.

3. Metric awareness is necessary in building a network of implications and expectations which lies at the heart of the musical experience. London [Lon92] proposes that metric cognition involves a two stage process comprising a recognition phase (establishment of a metrical framework) and a continuation phase (projection of the chosen framework into the future). Thus meter is critical in establishing expectations. London maintains that most computational and experimental studies of meter regard the recognition stage while theoretical studies provide retrospective evidence. Implied here is a failure to provide an adequate study of metric recognition that incorporates prediction and continuation. Our experiments take this challenge as a point of departure.

We propose that a listener simultaneously activates two parallel metric schemas each with some degree of independence. When one proves to correlate more consistently with other incoming patterns (dynamic accentuation, harmonic accent, phrase and articulation accents, etc.) the metric schema that fails 'turns off'. Furthermore, our model enables the integration of mutual influences of two interrelated aspects of musical expectations: schematic metric awareness (which influences functional tonal expectations) and learned functional tonal implications that in and of themselves create metric expectations. The merger and integration of these cognitive processes allow for a more refined model of music audition.

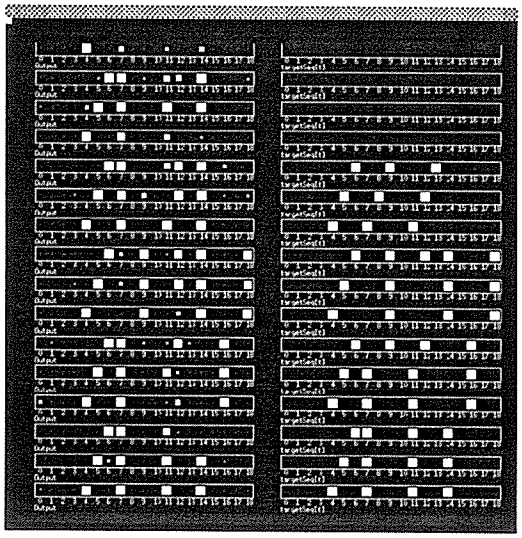

Figure 1: Simulation of expectancies. From left to right - 4 units represent duple meter and 3 more represent triple meter, the last 12 units are the harmonic expectations represented by 12 PCs. The size of the squares is proportional to the strength of the units' activity. Time proceeds from bottom-up. The right column represents the input and the left column visualizes the net's prediction for the meter and harmony. The progression is -
[3/4 I I I — vi vi ii — V V V7 — I I I]

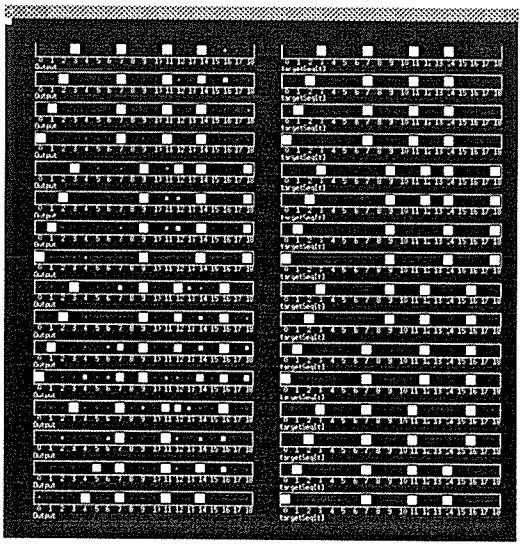

Figure 2: The progression is -
[4/4 I I vi vi —IV IV ii ii —V V V7 V7 —I I I I]

2 The network design

2.1 Architecture of the network

In our previous model of fluctuation in DRE (see [GB96] and [BG96]), we adopted a three-layer sequential net in which 12 state units establish the context of the current chord sequence, and the 12 output layer units represent the prediction of the net for the subsequent chord. Both, the state and the output units are pitch class (PC) representations of triads and tetrads in the sequence. The output layer is fed

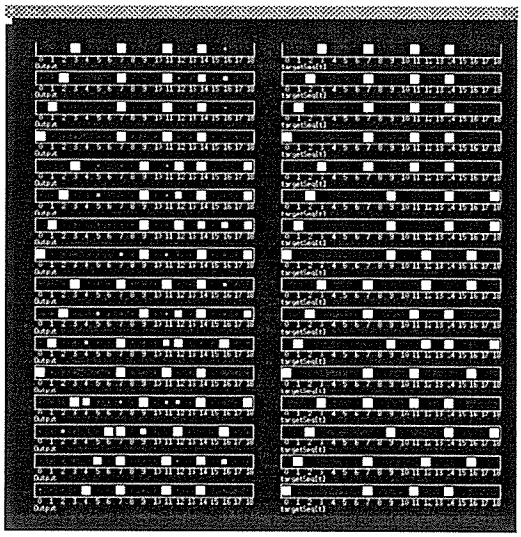

Figure 3: The progression is -
[4/4 I IV V I —vi V7 I IV —ii V V I —I I I I]

back into the state units to influence the next prediction of the net. The value of the state units at time t is the sum of its value at time $t-1$ multiplied by decay parameter and value of the output units at time $t-1$. By integrating a sub-net with the sequential net we supplemented the model with a simple metrical organizer that supplied a periodic beat stream of four beats per measure of duple harmonic progressions.

This model is extended by adding triple meter patterns to the architecture. In so doing we examine how metric expectations can influence the harmonic predictions and how the harmonic progression together with the context of the meter influence the prediction of meter.

This architecture differs from the previous model in a number of respects. The representation of meter is extended. We incorporate into the net's state units two pools (or a sub-net) of: 4 units to represent duple meter and 3 more to represent triple meter. These units are connected to the hidden layer together with the pool of the PCs representing the harmonic context. The hidden units are connected to the output layer. The output layer contains three pools of units: a pool for the harmonic expectations represented by the 12 PCs; and the two pools to represent expectations for duple and triple meter. The output of the 7 units of the meter is fed back into the corresponding pools of the state. The output of the prediction of the net for the harmonies' expectation were used to measure DRE and the target was fed into the PC units of the state, to establish the current harmonic context. We thus model the mutual influences of harmony on meter and meter on harmony. We note the enhancement of this method in quantifying the DRE. The DRE is also influenced by the metric expectations. This is particularly evident in (fig 4) where conflicting metric information greatly affected the DRE.

2.2 The set of learning examples

We use a learning set of functional tonal harmonic patterns. The patterns were evenly divided into duple and triple meter progressions. Harmonic rhythm in the learning set ranged from one chord per measure to one chord per beat, although the weighting was on one and two chord changes per measure for both duple and triple patterns.

3 Running the Net

3.1 The Learning Phase

For the learning phase the net was given thirty examples containing duple and triple patterns of harmonic progressions. After training, the net was able to reproduce the examples. We have tested the performance of the network with several different learning parameters. For example we found that for this task the net required relatively high value for the decay parameter.

3.2 The Generalization Phase

In this phase the net was given four new sequences. The target sequence was compared to the current harmonic and metric prediction of the net. The meter was fed back into the meter's pools of the state units and the target of the current harmonies was fed into the PC units. In analyzing the output we consider the distribution of the units' activation. By calculating how much of the target is present in the harmony pool of the output units, we were able to suggest a quantitative measurement of the DRE. The units of the meter pools in the output reflect duple and triple interpretations and clearly demonstrate conflicting metric and harmonic information.

4 Data Analysis

4.1 Figure 1:
[3/4 I I I — vi vi ii — V V V7 — I I I]

This example represents the output of a standard four measure progression in triple meter. The progression should show a high DRE. The role of the metric subnet is critical in the network's agility in detecting the correct harmonic rhythm by beat five. Of note is

the openness of the system to change on beat three (resulting from the inconclusive assistance of the metric sub-net). However the downbeat of measure two entrains the network by supposing a metric schema which fully conditions expectation for harmonic progression and change. Thus, in measure two the expectation for a subdominant harmony is progressively strengthened and the expectation for a change to the dominant is highly expected. (The inconclusive expectation for tonic continuance in the final measure is an artifact of 'padding' the example in order to incorporate longer progressions).

4.2 Figure 2:
[4/4 I I vi vi —IV IV ii ii —V V V7 V7 —I I I I]

In this example a harmonic progression in 4/4 with a high DRE is input as a target sequence. In this example the initial willingness for change on beat three (evident in the distribution of strength of PC7 to PC5 and PC9 representing an expectation for shift to the sub dominant) is immediately followed in beat four by an even stronger expectation for change to a subdominant. The lowest DRE in the entire progression occurs in beat five. Here, the downbeat is fully recognized as a point of harmonic shift, with a greater expectation for sub dominant harmony, but with an openness for a dominant downbeat. The arrival of a subdominant in correspondence to the metric downbeat sets a strong expectation for the completion of the progression.

4.3 Figure 3:
[4/4 I IV V I —vi V7 I IV —ii V V I —I I I I]

In this example a distinct conflict between harmonic rhythm and meter results in significant drops in DRE. The hastened harmonic rhythm (a chord already on the second beat, setting up a quarter note harmonic rhythm) is resisted in the output's expectation for continued subdominant harmony in beat 3. The arrival of a tonic on beat four of measure one throws both the metric counter and the harmonic expectations into flux. The drop in DRE is particularly interesting in that the distribution of expectations is not willy nilly but rather reflective of an ambiguity, in which conflicting functional regions (tonic/ dominant) are confused. This conflict persists until the final measure.

5 Discussion

Some basic questions regarding the perception of meter in tonal music are raised. Specifically:

1. How does a listener identify the meter, when hearing an unfamiliar work?
2. Is the process of metric cognition one of parallel or sequential testing? That is, do we consider multiple possible meters simultaneously, or do we test one and, failing to achieve a good 'fit', shift to another metric count?
3. What are the implications of these questions on our theory of musical expectations?

In our first experiment we extended the initial model by incorporating two parallel and independent counters for three beats and four beats. An experiment currently being considered is to commence with two parallel counters but shut one off when a strong correlation between a high DRE and one of the two pools in the metric sub-net is established. A second experiment under current consideration involves a change of data structure, such that multiple metric possibilities are reflected within a single counter.

References

[BG96] J. Berger and D. Gang. Modeling musical expectations: A neural network model of dynamic changes of expectation in the audition of functional tonal music. In *Proceedings of the Fourth International Conference on Music Perception and Cognition*, Montreal, Canada, 1996.

[DH89] P. Desain and H. Henkjan. The quantization of musical time: A connectionist approach. *Computer Music Jouranal (CMJ)*, 13(3), 1989.

[GB96] D. Gang and J. Berger. Modeling the degree of realized expectation in functional tonal music: A study of perceptual and cognitive modeling using neural networks. In *Proceedings of the International Computer Music Conference*, Hong Kong, 1996.

[LHL82] H. C. Longuet-Higgns and C. S. Lee. The perception of musical rhythms. *Perception*, 11:115–128, 1982.

[LJ83] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music.* Cambridge (MA): MIT Press, 1983.

[Lon92] J. London. The cognitive implications of a dynamic theory of meter. In *Proceedings of the Fourth International Conference on Music Perception and Cognition*, Pennsylvania, 1992.

[Mat39] J. Matheson. *Der Vollkommene Cappelmeister.* Hamburg: Christian Herold, 1739.

[Pov81] D. Povel. The internal representation of simple temporal patterns. *Journal of Experimental Psychology: Human Perception and Performance*, 7:3–18, 1981.

Statistical Pattern Recognition for Prediction of Solo Piano Performance

Chris Chafe
Center for Computer Research in Music and Acourstics
Music Department, Stanford University
cc@ccrma.stanford.edu

Abstract

The paper describes recent work in modeling human aspects of musical performance. Like speech, the exquisite precision of trained performance and mastery of an instrument does not lead to an exactly repeatable performed musical surface with respect to note timings and other parameters. The goal is to achieve sufficient modeling capabilities to predict some aspects of expressive performance of a score.

1 Introduction

The present approach attempts to capture the variety of ways a particular passage might be played by a single individual, so that a predicted performance can be defined from within a closed sphere of possibilities characteristic of that individual. Ultimately, artificial realizations might be produced by chaining together different combinations at the level of the musical phrase, or guiding in real time a synthetic or predicted performance.

A pianist was asked to make recordings (in Yamaha Disklavier MIDI data format) from a progression of rehearsals during preparation of Charles Ives' First Piano Sonata for a concert performance. The samples include repetitions of an excerpt from the same day as well as recordings over a period of months. Timing and key velocity data were analyzed using classical statistical feature comparison methods tuned to distinguish a variety of realizations. Chunks of data representing musical phrases were segmented from the recordings and form the basis of comparison.

Presently under study is a simulation system stocked with a comprehensive set of distinct musical interpretations which permits the model to create artificial performances. It is possible that such a system could eventually be guided in real time by a pianist's playing, such that the system is predicting ahead of an unfolding performance. Possible applications would include performance situations in which appreciable electronic delay (on the order of 100's of msec.) is musically problematic.

Caroline Palmer's comprehensive review of studies of expressive performance [1] presents several points that bear importance for the present work. Foremost, she warns against "drawing structural conclusions based on performance data averaged or normalized across tempi."
Data in the present work is analyzed in a way that preserves nuances until the final steps of classification.

Several reports are mentioned in conjuntion with the exploration of structure-expression relationships and corroborate the salience of phrase-level units in performance analysis. For example, errors in complex sequences when analyzed suggest that phrase structures influence mental partitioning. Errors tend not to interact across phrase boundaries. Also, phrases appear to be tied to their global context in different ways. Some phrases appear to be "tempo invariant" where others scale according to tempo-based ratios.

Palmer states, "Each performer has intentions to convey; the communicative content in music performance includes the performers' conceptual interpretation of the musical composition." Expressive

variations are intentional and show a high degree of repeatibiliy in patterns of timing and dynamics. Performers are deliberate in applying devices to portray their concepts, for example choosing louder dynamics to strengthen unexpected structural or melodic events. Events with higher tension (in a tension / relaxation scheme) might be brought out by being played longer.

2 Data from Rehearsals

Pianist George Barth, a Professor of Performance in the Stanford University Music Department, provided the recordings. He prepared his performance over the course of four months with nearly daily practice. The first five samples that are analyzed here were collected over several weeks, beginning after he felt confident of the notes.

An extract of the fifth movement was targeted for study after an initial look at the data confirmed good stability across the five samples. The 55 note passage was performed flawlessly in each take and provided sufficient length and variation for purposes of the analysis. The pianist was unaware of the the choice of the extract, so as far as he was concerned he was recording a much longer excerpt of the movement, thus avoiding any likelihood of study-influenced effect on the performance.

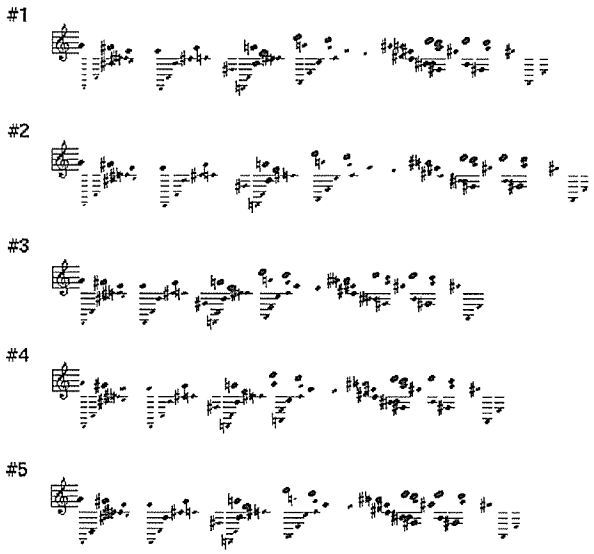

Figure 1: Displayed proportionally, the raw data for note onsets and key velocity shows expressive variations.

Several steps were necessary to prepare the extract for analysis. The performances were recorded directly to the Disklavier's floppy disk in Yamaha's E-Seq MIDI data format. Conversion to Standard MIDI File Format type 1 was accomplished in software with Giebler Enterprises' DOMSMF utility. Segmentation of the extract and conversion to type 0 format was accomplished with Opcode Systems' Vision sequencer. Trimmed and converted files were then imported into the Common Music Lisp environment for the first stages of analysis.

The present study is limited to note onset timings and key velocity (dynamic) information. Duration and pedaling data have been preserved during the conversion process for possible subsequent use.
Figure 1 is a proportional graph depicting the raw quantities recorded from the five perfomances. In Figure 2, phrase timing differences are highlighted by connecting a line segment between the positions of the starting and ending note-heads of each phrase.

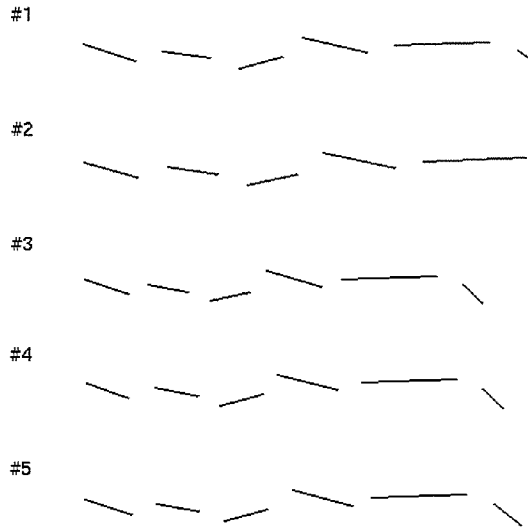

Figure 2: Sketching only phrase boundaries, tempo changes are visible both globally across phrases and internally within phrases.

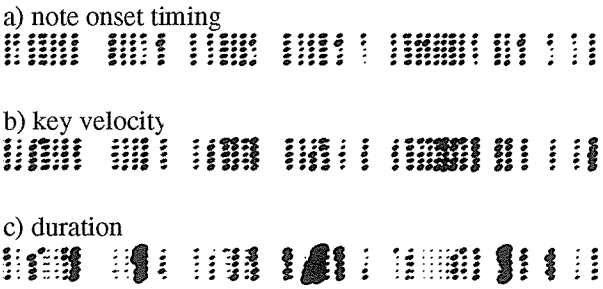

Figure 3: Variation in three parameters across the five performances.

For ease of comparison, Figure 3 isolates parameters with phrases aligned (by lining up events on the timings of the first performance and varying the notehead size according to the parameter). In b), variations of note onset timing use data relative to the first performance (larger noteheads indicate greater lengthening). Dynamic information is depicted by notehead sizes that depend on the key velocities found in each performance. Durational information is shown for informational purposes but was not analyzed further.

3 Covariance Analysis

Performance data, being sequential, requires the choice of a time window relevant to the features that the analysis intends to capture. As can be seen in the above graphs of the raw data, phrase-level comparisons are of interest. Phrases have different overall durations and begin times and are influenced by the tempo of the performance. The first step in preparing features for classification was to isolate the phrases, setting the elapsed time of each event to be relative to the onset of the phrase rather than its absolute time.

The two features chosen as dimensions for a covariance analysis are note onset timings and dynamics expressed as differences from a reference performance (key velocities are scaled to a range of 0 - 1). A less effective approach would be to express differences relative to perfect values derived from proportions in the score, which itself is a sort of performerless performance. Differences obtained against the score are distributed more coarsely; timings are relative to a less realistic baseline and values for dynamics have to be intuited (since they are specified only generally). By referencing to a recorded performance, differences

are distributed more usefully. Stylistic or habitual features such as phrase-final lengthenings are made implicit and dynamic differences are relative to actual values.

To compare two performances, three performances are required: the reference (P_{ref}) and the two inputs (P1 and P2). For each phrase, each event in each input is mapped according to the two feature dimensions. The intended result is that the inputs will be sufficiently distinguishable in this space. Figure 4 shows the distribution that results for the fifth phrase with Pref as performance #5, P1 as #1, and P2 as #2. A separator has been calculated based on the Mahalanobis distance to the center of each performance cluster [2]. The separator as shown correctly classifies 76% of the displayed points.

As the performance unfolds, the relative positions of cluster centers change phrase-by-phrase. Figure 5 shows trajectories mapped for four performances during the second half of the excerpt.

The analysis demonstrates an ability to correctly classify nearby performances. In Figure 6, a coincidentally close pair of performances for one phrase was correctly classified.

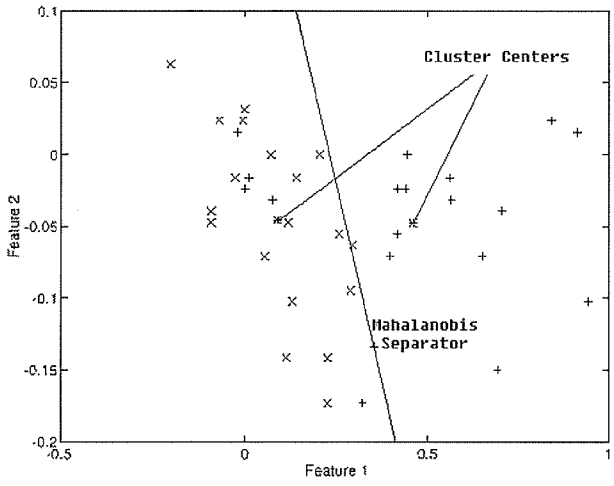

Figure 4: Note onset timing (feature 1) is plotted against key velocity (feature 2) for the same phrase in two performances. Quantities are differences from values for the same notes in a third, reference performance.

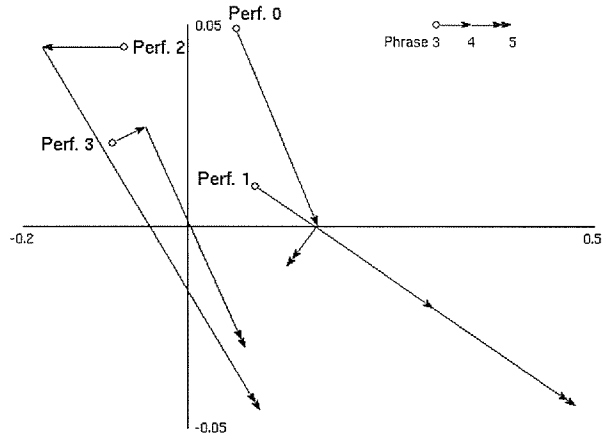

Figure 5: The relative positions of cluster centers change phrase-by-phrase. The trajectories of four performances are shown for three phrases in the same feature comparison space as Figure 4.

4 Discussion

Phrase-by-phrase tendencies in rhythmic and dynamic articulations can be successfully classified by covariance analysis. Performances that are not distinguishable are presumed similar for the sake of the

model being developed.

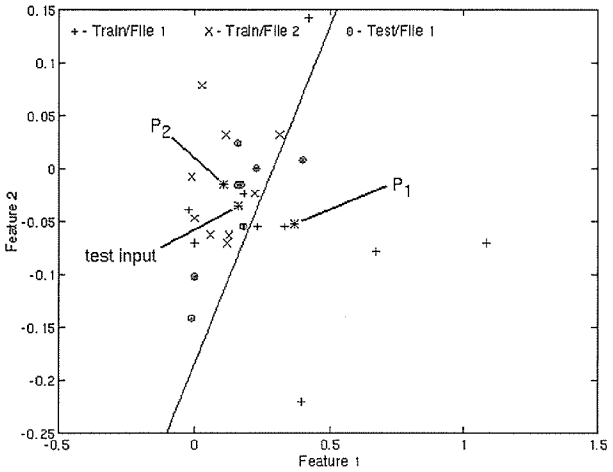

Figure 6: Successful classification of an "unknown" performance of phrase #4 in the comparison space of performances 1 and 3.

A future interest is to produce imitative expressive performances via behavior-based manipulation. A given passage would be realized by selecting a stored phrase from an analyzed set of phrases. In a purely guided mode, the operator would determine the sequence of phrase samples, perhaps also choosing from interpolated combinations as in [3]. Another mode involves real-time analysis / synthesis of expressive performance. A pianist performing in real time would be located in the comparison space and on-the-fly classification decisions would predict the most likely stored performance matching the current input. The ability to predict ahead of a current performance can be useful, for example to overcome transmission delays.

The predict-ahead capability is analgous to teleautonomous control in robotic applications [4]. The remote instrument (robot) is played by its predictor (a remote simulator) guided by controls transmitted to it by analysis of the local performer (human operator). To be agonizingly complete in this analogy, a remote accompanist's performance (environmental feedback) is provided back to the local performer via a second system running in the other direction. A bi-directional setup might allow a piano duo to perform together across oceans. The two simultaneous concerts would differ, but not by much, assuming the analyzers and predictors are effective.

A performance is made of many layers. Global tempo
changes and other longer structures remain to be described in the present model. Force-feedback manipulation of the model is discussed in O'Modhrain's accompanying article [5]. Her system operates on the phrase-level substrate that has been the focus of the present analysis and is intended to display the possible realizations of a given phrase within its comparison space. As a performance unfolds, the manipulator is guided through a dynamically changing scene, much like Figure 5.

Arkin describes layers of schema operating in combination to enable guided teleautonomous behavior of a robot. "...that schema-based reactive control results in a 'sea' of forces acting upon the robot." By patterning phrase-level behavior according to a predictor, partially autonomous performance is possible which can be realized in conjunction with global and other performance schema. Control of these other layers is a subject for future work, either in testing a real-time remote performance venue or in an editing environment for algorithmic performance.

5 Acknowledgments

Many thanks to George Barth for discussions and another round of experiments dissecting his excellent piano playing. Also to participants of the HCI Design Course Fall '96 at San Jose State, Stanford and Princeton Universities.

References

[1] Palmer, C. 1997. "Music Performance," Ann. Rev. Psychol., 48, pp. 115-38.
[2] Devroye, L., et al. 1996. *A Probabilistic Theory of Pattern Recognition*, New York: Springer-Verlag.
[3] Chafe, C., S. O'Modhrain 1996. "Musical Muscle Memory and the Haptic Display of Performance Nuance" Proc. ICMC, Hong Kong
[4] Arkin, R. 1991, "Reactive Control as a Substrate for Telerobotic Systems," IEEE AES Sys. Mag.
[5] O'Modhrain, S. 1997. "THE FUZZY MOOSE: A Haptic Tool for Tracking the performance of Fuzzy Classifiers in real-time.," Proc. ICMC, Thessaloniki.

Impact of MIDI on Electroacoustic Art Music

Alex Lane Igoudin
CCRMA, Department of Music, Stanford University
aledin@ccrma.Stanford.EDU

Abstract

The revolution in the tools for music composition in the mid-1980's caused a major change in electroacoustic music composition itself. Inherent features of the MIDI protocol and design of MIDI devices have had numerous implications for compositional practice. This paper highlights the reception, adaptation, and application of MIDI tools on a sampled researched group of electroacoustic art music composers and analyzes the impact MIDI has had on composition with electroacoustic media. The full text of the dissertation is available from CCRMA.

1 Introduction

MIDI was not the first technological revolution to affect art. However, few such revolutions brought change to the degree that MIDI did. Never before, nor ever since the introduction of MIDI, have we witnessed a change in the tools employed by the artists for the creation of art as quick, universal and profound as in the case of the standardization of digital musical instruments and computers in the mid-1980s. MIDI-based music technology provided an entirely new and comprehensive array of composition tools. The flood of MIDI-based hardware and software appearing within two years after the introduction of the standard transformed the concepts of the contemporary electronic music studio, the digital instrument, and the role the computer plays in musical composition.

Transformed but nonetheless present in electroacoustic music, are brought to the fore of the current research. Besides inducing the emergence of a new style in electroacoustic music, the MIDI phenomenon has also highlighted an array of thorny aesthetic and technological issues. Since the technology was not directly designed for the demands of non-commercial art, conflicts arose in the interaction between these two spheres. Thus, the anticipation of MIDI, the initial reaction to it, and the evolution of the reception of MIDI in the art music community are addressed in the study.

The results of this survey accurately reflect the attitudes and experiences of a sampled group of composers. However, the author is convinced that these results can also demonstrate the trends existing in the entire electroacoustic community.

2 Research Design

The study concentrates on the impact that MIDI has had on art music composition, focusing on composers who were active in *electroacoustic art music* before and after the introduction of MIDI. Due to insufficient bibliographic sources, the author chose to interview these composers as the method to obtain research data. Forty-five composers from thirteen countries including composers from both West and East Coasts of the United States were interviewed in 1996 using the same questionnaire. The collected interviews laid down the foundation of the study.

3 Introduction to MIDI

The majority of respondents – nearly two-thirds – did not experience problems arising from incompatibility of pre-MIDI hardware. General lack of communication and control between the pre-MIDI devices was habitually solved by custom-built or same-brand compatibility or, more often, by avoiding the use of mutually unacceptable hardware. Few composers (17%) attempted to forcefully connect incompatible elements of pre-MIDI setup.

Despite the persisting myth of anticipation of MIDI, 79% of those who answered the question "Did you expect the appearance of a control standard like MIDI in the mid-1980s?" responded negatively. The 'it-was-a-surprise' pattern dominated the answers.

MIDI and its foremost carrier, the Yamaha DX7, were introduced nearly simultaneously in different parts of the world. Even though manufacturers of Yamaha and other early MIDI synthesizers were pre-

dominantly based in Japan with some technology originating in California, the survey shows no correlation between the locality of the respondent and the time of introduction.

The absolute majority of the composers working in the field became aware of MIDI and MIDI devices within the first 2 years of its existence: 82% of the surveyed composers were introduced to MIDI in 1983-84. In 1985, the percentage grows to 91%.

The rate of applying MIDI to composition was much slower: compared to 82% who were introduced to MIDI, only 19% of the composers wrote pieces involving MIDI equipment in 1983-84. Only in the beginning of 1990s does application of MIDI tools finally catch up with their introduction.

There is a deep connection between introduction of both MIDI and Yamaha DX synthesizers in the communal memory. 19 out of 45 composers mentioned Yamaha DX7 answering the question of the time and place of their introduction to MIDI. Surprisingly, they mentioned this connection without being prompted to recall any such link.

4 Reception of MIDI

The speed at which the composer adapted to new tools depended on how quickly the coming MIDI gear answered his/her compositional needs. The initial impression of MIDI was often laid against the background of the long-term experience of working with pre-MIDI electroacoustic music, in particular, its achievements. What MIDI equipment could not do, but the analog or pre-MIDI digital could, was often the main source of hesitation for using it. Initial reaction to MIDI and MIDI equipment among the art composers can be divided into 2 roughly equal groups: 21 interviewees responded positively, showing interest and approval of the new technology; for remaining 24, the downsides of this technology prevailed in their initial opinion.

Surprisingly, half of the sampled population (51%) have changed their opinion about MIDI over the time MIDI tools have been in use. Overall, the breakdown of favorable/unfavorable opinion percentage evolved from 54:46 at the time of introduction to 80:20 currently. This is a remarkable evolution of attitude to MIDI technology in the surveyed group from roughly evenly divided in the beginning to the absolute majority favoring in the end.

The majority of the surveyed composers (68%) agreed that "MIDI tools were easy to learn, install, implement into the composition environment."

28 (76%) of 37 composers who answered the question attempted to transfer their pre-MIDI compositional methodology into their MIDI pieces. The transfer was successful with 20 out of 28 of these composers. The failure to transfer was present in the answers of the remaining 3 composers. The numbers above show that the majority of composers successfully transferred their pre-MIDI techniques into their MIDI pieces.

Of 45 electroacoustic composers interviewed, 30 (two-thirds of the entire sample) have used MIDI for their pieces on a regular basis. 11 cited use of MIDI on a minimal basis, peripheral to their pieces and/or, their use in only one piece. A good example of the latter use is a composer who used MIDI in only one piece, being quite prolific before and after.

The decision to use MIDI equipment has been determined by the balance between the assessment of its advantages to express compositional ideas and tolerance of its limitations. The use of MIDI instruments also depended on the aesthetic criteria of the composer and the availability of instruments. When asked if they continued to use exclusively non-MIDI environments after introduction to MIDI equipment, 63% (27 out of 43 composers) responded positively. Only 16 (roughly one third) showed complete transfer to MIDI tools.

5 Social Benefits of MIDI Equipment

Nearly 4 out of every 5 composers (79%) found the social access to the tools much better with MIDI than with pre-MIDI equipment. *Financial accessibility* is commonly cited as the main social benefit of the MIDI generation of music-making tools. *Democratization* of electroacoustic music (obviously only compared to pre-MIDI conditions) was another important social benefit of MIDI: the community has expanded as the access to the tools has been given to the musicians not associated with an institution including the musicians coming from underprivileged backgrounds and from non-Western musical traditions.

Commonality of MIDI devices due to their wide distribution was to the advantage of touring performers, particularly in troubleshooting equipment problems. Use of a MIDI device as a substitution for an acoustic instrument could also save the extra costs and rehearsal time when organizing a concert. *Portability* was another benefit for organizing concerts. Extended *performability* of MIDI-based pieces, due to equipment interchangeability, was another benefit to concert practice.

6 Timbre and MIDI

The ready-made timbres available in MIDI instruments, usually referred to as *MIDI presets*, opened a universe of new sounds and compositional solutions. For the first time, libraries of malleable, MIDI-controllable timbres became a major timbral source for composition. The very concept of offering a bank of believable sounds imitating acoustic instruments as well as some entirely original patches, caught on quickly with popular music, but did not fare as well in art music. Only about half of the interviewed composers ever used MIDI presets in composition. Three out of every four composers used software synthesis, but only two out of four would use MIDI timbres.

Only 31% of composers whose answers are available hold positive opinions about MIDI presets. The negative opinions were shared by twice as many respondents. However, among the actual MIDI users, presets were rated much better with 52% viewing them positively vs. 38% negatively.

The MIDI generation of electroacoustic equipment featured a new type of instrument – a controllable module with ready-made synthesized sounds. This signified a major turn in the concept of synthesizer: from a unit responsible exclusively for synthesis of sounds to a bank of sounds with limited synthetic facilities.

The voice-bank synthesizers had come into existence in the beginning of the 1980s. However, the idea of the ready-made sound itself contradicted with the contemporary approach which emphasized unlimited possibilities for experimentation. As a result, presets enjoyed little popularity before or with MIDI. Easy access to the libraries of ready-to-use sounds paradoxically created a new constraint – the limited number of those timbres leading to a lack of freshness or novelty in sounds. For composers who were used to the extreme range of available timbres that electroacoustic music provides, this constraint ruled out the use of MIDI tools. A number of interviewed composers did not use MIDI synthesizers because they were not interested in using a MIDI patch instead of an acoustic instrument.

The absolute majority of the presets in MIDI instruments simulate acoustic instruments. The general opinion about the quality of simulation in MIDI preset timbres is overwhelmingly (86%) negative. In the responses, several composers distinguish the degree of simulation success with different orchestra groups: percussive and plucked sounds fare better than others; simulated strings and trumpets did not receive many compliments. Software synthesis has continued to be the predominant source of sounds: only about a half of composers used MIDI presets but three-quarters used software synthesis after introduction to MIDI. In fact, the use of software synthesis in the sampled group has modestly grown from two-thirds of the respondents in pre-MIDI years, to three-fourths, during MIDI years. In only two cases has the use of MIDI timbres phased out the software synthesis.

7 Live Interactive Electroacoustic Music

There is a clear connection between the turn to writing live interactive pieces and the introduction of MIDI tools: with the coming of MIDI the percentage of composers writing music for live interactive performance grows from 41% in the pre-MIDI days to two-thirds. Of those who composed live music after their introduction to MIDI, 60% agreed to the statement that the availability of MIDI influenced their choice for writing live interactive music. Meanwhile the number of composers who wrote tape pieces slightly decreased (to 84%) as some switched entirely to live interactive music.

Half of the respondents in the survey reported using improvisation in their compositional practice. All composers who answered the question consider MIDI facilities for improvisation better than the ones available before MIDI equipment.

8 MIDI and Use of Notation

Despite the existence of many practices contradicting traditional conventions of Western art music, the absolute majority of interviewed electroacoustic composers (76%) acknowledge use of notation as part of their compositional method. More strikingly, 63% have used traditional notation for the composition of electroacoustic music. The MIDI note-oriented approach fit perfectly into note-oriented compositional practice: the input of notes was greatly simplified, and the response time for trying out the sketch minimal and user-friendly approach of contemporary digital equipment was helpful. In that regard the contribution of MIDI technology to the notation practices has been revolutionary. Forty-four percent of surveyed composers have used MIDI notation software. Answers show a balance of demands met and unfulfilled by MIDI **notation packages.**

9 Change of Compositional Interest

Only 8 composers (22%) have changed their focus of compositional interest in electroacoustic music from one property to another in transition from pre-MIDI to MIDI environments. Despite the profound change in the tools, the core of the compositional activity remains the same. The tenacity with which the composer's style persists is indirectly supported by the overwhelming number of composers who decided to transfer pre-MIDI practices into their MIDI works.

The eight composers who did switch demonstrated the shift of in interest to those aspects of composition that were greatly expanded by MIDI tools: interactivity, improvisation, and control.

10 Conclusions

The introduction of all-digital MIDI-based technology was not a mere change in the continuum of development in electroacoustic music technology, but much more a revolution in the tools used for composition. The extinction of analog devices came as one of the major effects of this revolution. The birth of new paradigms of instrument design, compositional environment and performance solutions were also significant.

The interaction between composers and MIDI tools is always a compromise between demands of the individual style and advantages and limitations of the MIDI equipment. Advantages and limitations of the protocol, further complicated by the implementation of MIDI and other technologies in the MIDI equipment, have had multiple effects facilitating or constraining the compositional process. In some cases the limitations of MIDI equipment and satisfaction of working with non-MIDI environments has led to the total exclusion of MIDI from the compositional setup.

The tenacity of tradition went against the drastic change of tools. The results show how the pre-MIDI genres and the timbral sources have continued to dominate in this period as well. The majority of art music composers have attempted to transfer their pre-MIDI compositional methodology and practices into their MIDI works. Most of them succeeded in this transfer. The switch to the new set of tools caused a change in compositional interest from one property of electroacoustic composition to another in less than a one-fourth of the surveyed composers; however, stylistic changes are numerous and vary from composer to composer. We see manifestations of that in all steps and elements of compositional process from organization of this process to changes in the structure of MIDI-based pieces. Distinct changes in the produced musical output appear as a result of such influence.

These results lead us to conclude that the adoption of new technology has not affected the core of the compositional style of the majority of composers, but caused abundant examples of changes to the details of methodology.

References

[1] Bouma, G. D.; Atkinson, G. 1995. *A Handbook of social science research.* Oxford : Oxford University Press.

[2] 1989. *On the wires of our nerves : the art of electroacoustic music.* Edited by R. J. Heifetz. Lewisburg : Bucknell University Press, London.

[3] Igoudin, A. 1997. *Impact of MIDI on Electroacoustic Art Music*, Ph.D. thesis, Music Dept., Stanford University.

[4] Loy, G. 1985. "Musicians Make a Standard: The MIDI Phenomenon", *Computer Music Journal*, Volume 9, Number 4, Cambridge, Mass. : MIT Press, pp.8-26.

[5] Manning, P. 1993. *Electronic and computer music.* 2nd ed., Oxford : Clarendon Press.

[6] 1990. *MIDI 1.0 Detailed Specification.* Los Angeles : International MIDI Association.

[7] Milano, D. 1984. "Turmoil in MIDI-Land", *Keyboard Magazine*, June 1984, pp. 42-106.

[8] Moore, F. R. 1988. "The Dysfunctions of MIDI", *Computer Music Journal*, Volume 12, Number 1, Cambridge, Mass. : MIT Press, pp. 19-28.

[9] 1989. *The Music machine : selected readings from Computer Music Journal.* Curtis Roads, Editor, Cambridge, Mass. : MIT Press.

[10] Rothstein, J. 1995. *MIDI : a comprehensive introduction.* The computer music and digital audio series, Volume 7. 2nd ed., Madison, WI : A-R Editions.

[11] Smith, D. 1993. "Dave Smith On MIDI's Early History", *Keyboard Magazine*, February 1993, pp. 70-86.

CCRMA Studio Report

Fernando Lopez Lezcano (nando@ccrma.stanford.edu)

Center for Computer Research in Music and Acoustics(CCRMA),
Stanford University

1. The place and the people

The Stanford Center for Computer Research in Music and Acoustics (CCRMA) is a multi-disciplinary facility where composers and researchers work together using computer-based technology both as an artistic medium and as a research tool. CCRMA is located on the Stanford University campus in a building that was refurbished in 1986 to meet its unique needs. The facility includes a large quadraphonic experimental space with adjoining control room/all digital studio, a recording studio with adjoining control room, a MIDI-based small systems studio, a general purpose analog/digital studio, several work areas with workstations, synthesizers and speakers, a seminar room, a reference library, classrooms and offices.

For a detailed tour and more information feel free to visit us in the World Wide Web:

- **http://ccrma-www.stanford.edu/**

The CCRMA community consists of administrative and technical staff, faculty, research associates, graduate research assistants, graduate and undergraduate students, visiting scholars and composers, and industrial associates. Departments actively represented at CCRMA include Music, Electrical Engineering, Mechanical Engineering, Computer Science, and Psychology. CCRMA has developed close ties with the Center for Computer Assisted Research in the Humanities (CCARH), recently affiliated with the Department of Music.

Staff & Faculty: **Chris Chafe**-Associate Professor of Music, Director; **Jay Kadis**-Audio Engineer/Lecturer; **Fernando Lopez-Lezcano**-Systems Administrator/Lecturer; **Heidi Kugler**-Secretary; **Max Mathews**-Professor of Music (Research); **Jonathan Berger**-Associate Professor of Music; **Julius Smith**-Associate Professor of Music and Electrical Engineering; **John Chowning**-Professor of Music, Emeritus; **Leland Smith**-Professor of Music, Emeritus; **John Pierce**-Visiting Professor of Music, Emeritus; **Jonathan Harvey**-Professor of Music; **David Soley**-Assistant Professor of Music; **Eleanor Selfridge-Field**-Consulting Professor of Music; **Walter Hewlett**-Consulting Professor of Music; **William Schottstaedt**-Research Associate; **Dan Levitin**, Lecturer; **Marina Bossi**; Lecturer.

2. The activities

Center activities include academic courses, seminars, small interest group meetings, spring and summer workshops, and colloquia. Concerts of computer music are presented several times each year including an annual outdoor computer music festival in July. In-house technical reports and recordings are available, and public demonstrations of ongoing work at CCRMA are held periodically.

Research results are published and presented at professional meetings, international conferences and in established journals including the Computer Music Journal, Journal of the Audio Engineering Society, and the Journal of the Acoustical Society of America. Compositions are presented in new music festivals and radio broadcasts throughout the world and have been recorded on cassette, LP, and compact disk.

3. The environment

The computing environment currently supported includes Macintosh computers and several flavors of unix-based workstations. The old and trusty network of NeXT computers has been augmented by two new supported hardware and software platforms. High powered Pentium and PentiumPro PC's are running both NEXTSTEP and Linux, the last one a fairly recent addition to the supported operating systems list. A couple of SGI machines running Irix complete the current setup. Several servers offer shared resources that are available in all platforms, an Ethernet network being the glue that ties everything together and connects CCRMA to the rest of the Internet. Supported software in the unix world includes the CCRMA Lisp Environment (which includes Common Music, Common Lisp Music and Common Music Notation), the MusicKit and associated programs (in NEXTSTEP only) and tons of utilities and packages for research and music creation. The Macintosh world has a full complement of MIDI based tools and is mostly used for MIDI applications, notation and digital mixing (with hardware assist from Dyaxis II and ProTools systems in two of the studios).

MIDI-based systems include Yamaha, Roland and Korg equipment including Yamaha DX, TX, SY, TG and VL

synthesizers, KX88 keyboard controller, Disklaviers, Korg WaveStations and Wavedrum, E-mu samplers and digital delays and reverberation. Also available are IVL pitch trackers, a Buchla Lightning MIDI controller, several Mathews Radio Drum controllers, MIDI patchers and drum machines from Yamaha and Roland.

Studio recording equipment includes a 24 track mixer, an 8 track TEAC analog recorder, a Yamaha DMR8 digital recorder and mixing console, several TEAC 8 track digital recorders, various signal processing devices, Westlake monitor speakers and an assortment of high quality microphones.

4. The research

This array of brief research summaries will give you an idea of the current crop of research at CCRMA and who's doing it:

Computer Music Hardware and Software:

- "PadMaster, an Interactive Performance Environment. Algorithms and Alternative Controllers" - **Fernando Lopez Lezcano**
- "Common Lisp Music and Common Music Notation", "The *snd* Sound Editor" - **William Schottstaedt**
- "The CCRMA Music Kit and DSP Tools Distribution" - **David Jaffe** and **Julius Smith**

Physical Modeling and Digital Signal Processing:

- "Flaring Bores" - **Dave Berners**
- "Reducing Numerical Computation in Struck String Physical Models" - **Stephan Bilbao**
- "Simple but powerful extensions to sample playback synthesis" - **Nicky Hind**
- "Oversampled Representations for Audio Parameter Estimation" - **Scott Levine**
- "ATS (Analysis, Transformation, Synthesis): source/filter (subtractive) algorithm and harmonic partials tracker design" - **Juan Carlos Pampin**
- String and wind synthesis, course development, filter design, numerous collaborations, First Tesseract CD finished and available (http://www.till.com/tesseract/) - **Julius Smith, Assoc. Prof. Music and EE**
- Reducing aliasing in Virtual Analog synthesis, and applying control-systems techniques to acoustics and physical modelling - **Tim Stilson**
- Physical modeling and non-linear acoustics - **Laurent Daudet and Julius Smith**
- "An Acoustic Analysis of Single-Reed Woodwind Instruments with an Emphasis on Design and Performance Issues and Digital Waveguide Modeling Techniques" - **Gary Scavone**
- "Transient Modeling Synthesis: a flexible analysis/synthesis tool for transient signals" - **Tony S. Verma, Scott N. Levine, Teresa H.Y. Meng**

Controllers for Computers and Musical Instruments:

- "The Hummer Project: Developing a MIDI controller for people with disabilities" - **Andrew Einaudi, Neil Scott**
- "Improvisation program on the radio-baton" - **Max Mathews**
- "THE FUZZY MOOSE: A Haptic Tool for Tracking the performance of Fuzzy Classifiers in real-time" - **Sile O'Modhrain**
- "Virtual violin human-computer interface" - **Charles Nichols**
- "Solo Acoustic Guitar Music Reproduced on a Real-time Controller: Creating a Better General Keyboard Controller" - **Jonathan Norton**

Psychoacoustics and Cognitive Psychology:

- "Denoising and Reductionism: Signal Analysis and High Level Reductive Analysis of Music" - **Jonathan Berger**
- "A Theory of Musical Expectation" - **Jonathan Berger**
- "Connected to What?" (Cognition in Music Theory) (SMT Plenary - 1997) - **Jonathan Berger**
- "A Neural Network Model of Metric Perception and Cognition in the Audition of Functional Tonal Music" - **Jonathan Berger and Dan Gang**
- "Statistical Pattern Recognition for Prediction of Solo Piano Performance" - **Chris Chafe**
- "Skill Development in Classroom Harmony and Keyboard Harmony: Breadth-First vs. Depth-First Learning" - **Paul von Hippel**
- "The Case for a Sequencer that Teaches Dictation Skills: Curricular and Flexibility Shortcomings of Existing Products" - **Paul von Hippel**
- "Impact of MIDI and MIDI Equipment on Electroacoustic Art Music" - **Alex Igoudin**
- "Automatic Pronunciation Scoring of Specific Phone Segments in Speech" - **Yoon Kim**
- "Composition and Collage: Morton Subotnick's *The Key to Songs*" - **Leigh VanHandel**

Computer Music and Humanities:

- "On Collaborations, Documents and Talking to a TV, Paper for the 1997 Symposium on Science / Art - Internet / Multi Media" - **Chris Chafe**
- "Musical Acoustics Research Library updates" - **Gary Scavone**
- "Ethnomusicological research in South America and report on the state of music technology in Chile and Argentina" - **Jane Rivera**

5. The music

Two CD's - "Computer Music @ CCRMA, Volume One" and "Computer Music @ CCRMA, Volume Two" - have been edited at CCRMA, in what is expected will be a continuing series representing the musical production of the center. Feel free to visit out web site for more details (http://www-ccrma.stanford.edu/).

Some of the recent (during this past year) compositional works realized at CCRMA:

- **Celso Aguiar** (DMA Graduate Student) - *All blue, I write with a blue pencil, on a blue sky*, for quad & stereo playback, *Sextet*, for flute, clarinet, percussion, piano, violin, cello.
- **Jonathan Berger** (Associate Professor of Music)- *The Voice Within a Hammer* (1), for flute, clarinet (Bb), violin, viola, piano and computer (ICMC 97), *The Voice Within a Hammer* (2), for flute, clarinet (Bb), mallet percussion, contrabass piano and computer, *Concerto for Piano and Orchestra.*
- **Joanne D. Carey** (Visiting Composer) - worked on improvisation programs in C for the Radio-baton for a piece in progress for Flute and Radio-baton.
- **Kui Dong** (Visiting Composer, China) - *Youlan: Long Winding Valley*, for stereo tape.
- **Janet Dunbar** (DMA Graduate Student) - *Song of the Shaman*, for performance poet, soprano, percussionist and stereo tape.
- **Gerald Eckert** (Visiting Composer, Germany) - Currently working on a tape piece, with a motet of Josquin Deprez as the basic sound material and structural meaning, *"Nen"* for clarinet solo, *"wie Wolken um die Zeiten legt"* for small ensemble, *"l'etendue des fins eclats, eparse"* for violin solo.
- **Jonathan Harvey** (Professor of Music)- **Juan Pampin** (PhD Graduate Student) **(collaboration)** - *Rumi* (tentative title), for choir and electronic sounds.
- **Nicky Hind** (DMA Graduate Student) *COSMOS*, for live electronics - using analog and FM synthesizers, and sampler, controlled by MAX and the radio baton.
- **Jun Kim** (DMA Graduate Student) - *ZephyrBells*, for quadraphonic sound, *DREAMING* for viola and computer-generated tape, *Reverberation*, for two sopranos, percussion, tape and five candlelights.
- **David Jaffe** (Visiting Composer / Researcher) - *The Seven Wonders of the Ancient World*, for Mathews/Boie Radio Drum-controlled Disklavier, mandolin, guitar, harp, harpsichord, harmonium, contrabass and 2 percussionists; Radio-Drum part is performed by Andrew Schloss, who also helped develop and refine it.
- **Tobias Kunze** (DMA Graduate Student) - "Protozoo, interactive sound installation".
- **Peer Landa** (Visiting Composer / Norway) - *Gag Order* (compact disk) This piece was commissioned by NoTAM for the GRM Acousmonium. The material is derived solely from three old native Japanese instruments and then rigidly processed by custom made DSP-applications. *Downcast* for tape using original C-based software.
- **Bobby Lombardi** (DMA Graduate Student) - performance of *"do you love me?"* for percussion narrator and tape, currently working on *"all you need"* for solo tape.
- **Fernando Lopez Lezcano** (System Administrator / Lecturer)- *With Room to Grow*, for PadMaster, Radio Drum, and MIDI instruments; *House of Mirrors* for PadMaster, Radio Drum, midi instruments and soundfile playback.
- **Charles Nichols** (PhD Graduate Student) - *interpose*, for guitar and computer generated tape.
- **Jonathan Norton** (PhD Graduate Student) - *Snapshots on a Circle* - for alto sax, cello, percussion and tape.
- **Juan Pampin** (PhD Graduate Student)- *Metal Hurlant*, for metallic percussion and electronic sounds, *Reverberation*, for two sopranos, percussion and computer processed sounds on tape.
- **Fiammetta Pasi** (Visiting Composer / Italy) - *Collage*, for stereo tape, *Quimeras*, for stereo tape.
- **Andre Serre** (Visiting Composer, France) work in progress for cello and four channel tape.
- **Kotoka Suzuki** (DMA Graduate Student)- *Eclipse*, for stereo tape and dance.
- **Marco Trevisani** (Visiting Composer / Italy) *Variazioni e Frammenti su Aura*, a Bruno Maderna inspired tape composition.Aura is an Orchestra piece written by Bruno Maderna in 1971. Signal processing, using CLM (Common Lisp Music)

Feel the Music: Narration in Touch and Sound

Maura Sile O'Modhrain
Center for Computer Research in Music And Acoustics
Stanford University
sile@ccrma.stanford.edu
http://www.ccrma.stanford.edu/~sile

Abstract

We describe the development of tools which allow us to track the relationship between multiple performances of a piece. Based on certain premises derived from Gestalt thinking, we develop a model for the relationship between music and haptics which allows us to explore the idea of a "narrative" element for haptic display. We report on preliminary work and suggest future directions.

1 Introduction

Any study which attempts to analyse musical performance must begin by tackling one knotty problem — how to define what is *performance* and what is *piece*. Mechanical instruments and computers have enabled us to render, more accurately than is humanly possible, the element of music that is *the piece*. What we seek now is a way of rendering the *not-the-piece*, the element which we shall hereafter refer to as "the performance."

Why are we interested in separating the two? The ability to independently control those elements which define performance opens up several exciting possibilities. Firstly, we can build a new class of instruments which already know how to read and play scores and which simply allow a player to control performance [4]. Secondly, we can make performance-related handles available during the process of digital music editing [2]. Thirdly, we can use this data to drive prediction algorithms to compensate for communication link timing lags and drop-outs in distributed rehearsal situations [1].

This paper introduces an altogether new approach to interpreting the element of not-the-piece, proposing for performance a model built on the idea of "narrative." Based on premises derived from Gestalt thinking, we propose a new modality for the presentation of performance-related data — the sense of touch and motion (called haptics).

Imagine your hand being guided by a puck through a space as you listen to a performance of a piece. In that space, compass points represent 8 past performances. As you hear the new performance, its closeness to each of these past performances is represented by the puck pulling your hand toward that compass point.

Below we describe our development of just such a display. We can provide haptic feedback or interactive forces concurrently with audio playback, thereby creating a tool for the analysis of musical performance and performance measures.

The question in designing this or any virtual environment which uses haptic display to convey time-varying information is how to glue objects or sequences of guided movements together. What we seek is a haptic equivalent to telling a story, a haptic narrative. Consistent with ideas in Gestalt psychology, especially those put forward by Gibson [3], impressions of the outside world can be essentially amodal, not associated with a particular sense. Most objects in the environment give rise to multimodal experiences. Our memory of a story is not dependent upon whether we read or heard it first. What we are endeavoring to discover in this present study is to what extent we can exploit our ability to abstract information from its mode of representation. Can we create a new haptic interaction with a piece that has nothing to do with how it is played physically but instead tells us something else about its performance?

At the heart of our work then is a hypothesis that, like real objects, pieces of music can have an internal representation which is independent of the senses by which they are first perceived. Further, since both haptic and audio information are gathered sequentially, they must share cognitive processes for constructing hypotheses based on the perception of events which unfold over a period of time.

In the sections which follow, we will briefly discuss previous work which has enabled us to develop our present hypotheses. Some theories from the field of cognitive psychology will provide us with a framework within which our concept of narrative can be developed. Finally, we will show that this concept of narrative can form the basis of a haptic display which will allow us to design experiments to test the hypothesis using data about past performances.

2 Background

In a previous paper [2], we describe a method for displaying music to the haptic senses, the senses of Taction and Kinesthesia, which projects timing and amplitude information about two performances onto a virtual wall. The wall's apparent stiffness is modulated by a parameter derived from note-onset and note velocity data obtained from two performers playing the same piece. By pressing against this wall, it is possible to build up an impression of the way in which local note groups and even more large-scale phrases were articulated by each performer. The success of this experiment lay primarily in the discovery that we could directly map the performer's manipulation of musical tension to the tension or stiffness of our virtual wall. The building and relaxation of musical tension was directly related to the changes in compliance of the wall over time.

Leveraging off this work, we began to build on two discoveries. Firstly, we had found that, for piano music at least, we could control some elements of performance which are independent of the score but which are consistent for each player. Secondly, we realized that what we had essentially done was to substitute one haptic narrative for another. We had replaced the haptic feedback from the piano with feedback from another instrument: one whose feel had no direct correlation to piano technique. We further realised that we now had a way of displaying phrase articulation to the haptic senses which required no knowledge of the feel of the instrument playing the music, and we began to develop the hypothesis that there could exist a thread of haptic narrative which could be exploited in the design of a new instrument. Like Max Matthews' Radio Baton [4], it would require no knowledge of instrumental technique. But unlike the Radio Baton which allows you to control the narrative element of MIDI playback by translating gestures into timing and loudness controls, it would take your hand on a tour of past performances, turning timing and loudness variations back into physical gestures.

3 Internal Representation

Central to our hypothesis is the idea put forward by Gibson that we build up an impression of the world around us by combining information gleaned from many senses [3]. A flower, for instance, is not simply defined by how it looks — it has a texture and a scent which are as much a part of its identity as its colour. White [5] took this one step further. He claimed that we build from abstract perceptual cues a model of our environment and, even though this model is a stylized and simplified abstraction, we believe it represents the truth. White called this tendency "distal attribution."

Now suppose we take these ideas and apply them to music. We have a single musical object, the piece. We can come into contact with a piece in many ways — we can hear it, look at a score, learn to play it, and so on. It is always the same piece, even when it is being whistled by a passer-by on the street. It possesses an integrity independent of its representation. We have built an internal mental model which may or may not correspond to how we first experienced the piece.

We can also build a hierarchical model with the piece in its most abstracted form at its top. Below this are various representations — the score, a recording, the musician's internal model of the piece as they learned to play it, and the listener's impression of a performance. Even further down this tree, branching from the performer, there are the various internal representations they have used to enable them to recall the music: visual memory (memory for the appearance of the score), kinesthetic memory (the memory of how the piece should feel to play), musical memory (an internal audio representation of the music), and maybe some theoretical framework which has enabled the performer to come to an understanding of the piece's structure. When performing, the musician must draw on some form of internal representation of the music, but which representation they use is not clear; probably they do not know. Whatever representation they use, their aim in performance is the same - to tell the story of the music based on their understanding of the piece to date.

What this hierarchy illustrates is that, no matter on which level you experience a piece of music, you take away an impression of the musical object that is independent of your point of contact with the piece. Listeners and players both build internal models which tell a story and it is, we propose, this narrative path through the piece which forms the basis for the independent representation of the musical object.

4 Displaying the Narrative

In using haptic display to convey information about performance, we take advantage of a pre-existing connection between haptics and music: the performer/instrument interaction. Most instruments require the player to come into physical contact with a mechanism of some sort, thus the translation of gesture into sound involves two sensory modalities — the musician perceives the sound of the instrument but also relies on its mechanical response for information regarding the results of their actions. An important component of playing music therefore is the continuous interaction between player and instrument via the haptic senses.

One way to think of a person's interaction with our display is that, while they are performing tasks, they are being performers and should be provided with the kinds of sensory cues that pertain. When they are exploring a space, however, they need different cues for remembering where they have been — just as the listener to a performance takes away a very different map of the piece they have heard from that which the performer has built for themselves. Both models rely heavily on past experience and both have strong narrative elements - the performer tells a story, the listener interprets it. How much the listener remembers and what in particular they are able to recall is of great interest here because it will determine how successful the performer has been in telling their tale.

How does this narrative element relate to our display? Imagine three performances of a piece. The first is a playback of a MIDI file containing exact note and timing data with constant key velocity. The second is an *In your face* interpretation where the performer has one idea which they forcefully project — like bringing out the top note of each arpeggiated figure in the piece by playing it louder. In the third performance, the performer wishes to engage the listener by telling a story - they articulate phrase structure and add tiny amounts of temporal variation which continuously force the listener to change their hypothesis about where the piece is going and to become involved in the story the performer is telling.

What is the equivalent in designing a haptic space for exploration? One could say that the computerized rendering of the piece is like dropping someone in a haptic space and leaving them to search it and build up a mental model for themselves of its features, somewhat like exploring a sculpture gallery where art works have no labels and there are no guidebooks. The *In your face* performance might be like taking the user around the space in a predetermined path which does not leave them room for any exploratory interaction, somewhat like being rapidly shepherded through the art gallery by a tour guide. The *engaging* performance might be like bringing someone into a haptic space and guiding them to particular *sites of interest* but then allowing them to explore these at their own speed, always being free to integrate them with the rest of the haptic environment - providing them with a guide book to explore the art gallery and labeling things clearly.

In other words, we would ideally seek to extract the narrative-defining elements from a performance and convert them into guided motions of our haptic puck. If we are successful, the audience, in this case the person holding the puck, should come away with a story that tells them about which of a set of past performances the one they are listening to is most like, for any point in time. What we offer is a programmable relationship between haptic objects and musical objects. This relationship is possible because both share a cognitive process — the creation of a narrative to connect sequential events to form hypotheses about their relationship to each other.

5 Our Haptic Display

Our haptic display device is the Moose, a two-degree-of-freedom planar device developed at CCRMA by Brent Gillespie. The Moose is comprised of two linear voice-coil motors connected to a puck or manipulandum by two perpendicularly oriented double flexures. The puck's position is tracked by two linear encoders and the whole is interfaced with a Pentium via a simple Digital I/O card. As the user moves the puck, forces can be exerted on their hand by the motors.

The playback of music is achieved through MIDI. MIDI data is output via a MIDI interface to a Yamaha Disklavier in real-time. It is worth noting here that, by using Haptic display to convey information about past performances we take advantage of a second channel of information exchange leaving the auditory channel available to monitor the *new* performance being played through the system.

6 Software

Our haptic environment runs under DOS and is interrupt driven which ensures that position sensing and force output are constant. We are also able to leverage off this accurate time-keeping to precisely co-ordinate output of forces to the Moose with the output of MIDI data to the Disklavier.

Our software, written in C++, draws upon a previously developed library of haptic objects to represent components of our data.

Our environment has two distinct modes. These are: 1) Tracking mode. Here a *new* performance is tracked continuously as it plays. Musical information obtained via MIDI [1] is transmitted via the MIDI interface to the Disklavier. At the same time, vectors obtained from the statistical classifier which determine the closeness of this performance to any previous performance are displayed as forces on the Moose puck which drag the user's hand toward that performance.

2) Exploration mode: Here the user's hand is no longer guided, but is free to explore the two-dimensional workspace of the Moose and to feel where each performance resides. Past performances are represented by "poles", virtual pillars which are located on the circumference of a circle.

By touching one of the pillars it is possible to audition its associated performance. Furthermore, the virtual springs which link each past performance to the new performance and cause the puck to be pulled around the workspace in tracking mode are tangible as grooves running from each performance location to the place where the puck stopped. These links can be broken, if desired, preventing their associated performance from influencing the puck.

The transition between modes is a simple toggle which acts like a pause control, taking up where it left off when mode is switched back to tracking.

A further advantage in using haptic output is our ability to exploit certain binarisms which are common to both our data representation and our haptic senses. The vectors which we receive as the output of our classifier describe how close or distant two performances are from each other within their parameter space. Since closeness and distance have direct correlates in haptic perception, we are able to take advantage of a binarism which exists in two modes — we can move toward and away from a performance in our haptic workspace.

7 Conclusions and Future Directions

We have derived from the principals of Gestalt thinking, from the principals of amodal representation and distal attribution, the concept of a piece of music that can exist as an object independent of its mode of representation. Using this concept, we have developed the hypothesis that pieces can exist as representation-independent objects because our interpretation and understanding of how they work is based upon constructing a narrative to connect their components as sequences of events in time. The performance, the not-the-piece element of music, is, we have proposed, the primary narrative-bearing element in realising a musical score. Analysing performance, therefore, depends upon being able to access this narrative element.

One way of accessing narrative is to display performance data to a sensory modality which shares with music cognitive processes for interpreting sequences of events in time. The modality we have presented here which meets these conditions is that of Haptic Display and we have presented preliminary work on a system for performance analysis based on this technology. Our primary objective now is to design some simple experiments which will allow us to discover to what extent musical and haptic narrative can interact. We are interested in discovering, for example, how we memorise music. In particular, what is the cross-modal interaction which allows us to recall both what a piece sounded like and what it felt like to play. We feel that understanding how musicians learn may provide some insight into haptic memory which will one day be of use to instrument designers and designers of virtual environments.

8 Acknowledgments

The author wishes to acknowledge the contributions of Brent Gillespie to this work.

References

[1] Chafe, Chris. 1997. "Statistical Pattern Recognition for Prediction of Solo Piano Performance" Proc. ICMC, Thessaloniki.

[2] Chafe, Chris, and O'Modhrain, Sile. 1996. "Musical Muscle Memory and the Haptic Display of Performance Nuance." Proceedings of ICMC, Hong Kong. pp. 428-431.

[3] Gibson, J. J. 1962. "Observations on Active Touch," Psychological Review. Vol. 69, pp. 477-491.

[4] Mathews, Max and Pierce, John. eds. 1989. Current Directions in Computer Music Research, MIT Press, Cambridge, Massachusetts.

[5] White, B. W, Saunders, F. A., Scadden, L., Bach-y-Rita, P., and Collins, C. C. 1970. "Seeing with the skin.", Perception and Psychophysics, vol. 7, pp. 23-27.

To be presented at the 1997 International Computer Music Conference, Thessaloniki, Greece.

Digital Waveguide Modeling of Woodwind Toneholes

Gary P. Scavone
gary@ccrma.stanford.edu

Julius O. Smith, III
jos@ccrma.stanford.edu

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305 USA

Abstract

This paper demonstrates a digital waveguide implementation of a six-hole woodwind tonehole lattice, based on theory and measurements published by Keefe (1981). Woodwind tonehole transmission-matrix parameters are converted to traveling-wave scattering parameters suitable for digital waveguide implementation. Second-order digital filters are designed to approximate the reflection and transmission transfer functions implied by the Keefe data. In this way, the tonehole is implemented by a two-port scattering junction which accounts for both series and shunt complex impedances. Alternatively, the tonehole can be implemented as a one-multiply, one-filter, three-port scattering junction. The results of a digital waveguide six-hole flute bore implementation using both models are compared to Keefe (1990), with excellent agreement. In this way, the best available acoustic theory regarding toneholes is efficiently and accurately simulated in discrete-time.

1 Two-Port Tonehole Model

The fundamental acoustic properties of toneholes have been extensively studied and reported by Keefe (1981, 1990). The model described by Keefe (1990) is an accurate representation for a tonehole unit, assuming adjacent tonehole interactions are negligible. In this description, acoustic variables at the tonehole junction are related by a transmission matrix of series and shunt impedance parameters. Keefe's original derivation of the tonehole parameters was based on a symmetric T section, as shown in Fig. 1 (Keefe, 1981). The series impedance terms, Z_a, result from

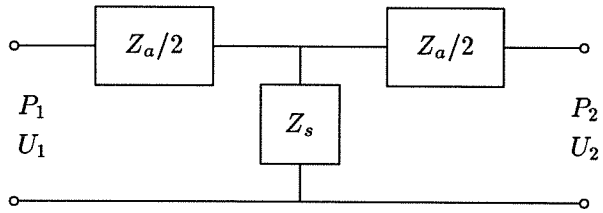

Fig. 1: T section transmission-line representation of the tonehole.

an analysis of anti-symmetric pressure distribution, or a pressure node, at the tonehole junction. In this case, volume flow is symmetric and equal across the junction. The shunt impedance term, Z_s, results from an analysis of symmetric pressure distribution, or a pressure anti-node, at the tonehole, for which pressure is symmetric and equal across the junction. The transmission matrix which results under this analysis is given by

$$\begin{bmatrix} P_1 \\ U_1 \end{bmatrix} = \begin{bmatrix} 1+\frac{Z_a}{2Z_s} & Z_a\left(1+\frac{Z_a}{4Z_s}\right) \\ Z_s^{-1} & 1+\frac{Z_a}{2Z_s} \end{bmatrix} \begin{bmatrix} P_2 \\ U_2 \end{bmatrix}, \quad (1)$$

obtained by cascading the three matrices which correspond to the three impedance terms. Based on the approximation that $|Z_a/Z_s| \ll 1$, Eq. (1) can be reduced to the form

$$\begin{bmatrix} P_1 \\ U_1 \end{bmatrix} = \begin{bmatrix} 1 & Z_a \\ Z_s^{-1} & 1 \end{bmatrix} \begin{bmatrix} P_2 \\ U_2 \end{bmatrix}, \quad (2)$$

which is the basic tonehole unit cell given by Keefe for transmission-matrix calculations. The values of Z_a and Z_s vary according to whether the tonehole is open (o) or closed (c) as

$$Z_s^{(o)} = Z_0(a/b)^2\,(jkt_e + \xi_e), \quad (3a)$$
$$Z_s^{(c)} = -jZ_0(a/b)^2\cot(kt), \quad (3b)$$
$$Z_a^{(o)} = -jZ_0(a/b)^2kt_a^{(o)}, \quad (3c)$$
$$Z_a^{(c)} = -jZ_0(a/b)^2kt_a^{(c)}. \quad (3d)$$

Definitions and descriptions of the various parameters in Eqs. (3a) – (3d) can be found in (Keefe, 1990).

To render these relationships in the digital waveguide domain, it is necessary to transform the plane-wave physical variables of pressure and volume velocity to traveling-wave variables as

$$\begin{bmatrix} P_1 \\ U_1 \end{bmatrix} = \begin{bmatrix} P_1^+ + P_1^- \\ Z_0^{-1}\,(P_1^+ - P_1^-) \end{bmatrix}, \qquad (4)$$

where Z_0 is the characteristic impedance of the cylindrical bore, which is equal on both sides of the tonehole. Waveguide pressure variables on both sides of the tonehole are then related by

$$\begin{bmatrix} P_1^- \\ P_2^+ \end{bmatrix} = \begin{bmatrix} \mathcal{R}^- & \mathcal{T}^- \\ \mathcal{T}^+ & \mathcal{R}^+ \end{bmatrix} \begin{bmatrix} P_1^+ \\ P_2^- \end{bmatrix}, \qquad (5)$$

where

$$\mathcal{R}^- = \mathcal{R}^+ \approx \frac{Z_a Z_s - Z_0^2}{Z_a Z_s + 2Z_0 Z_s + Z_0^2}, \qquad (6a)$$

$$\mathcal{T}^- = \mathcal{T}^+ \approx \frac{2Z_0 Z_s}{Z_a Z_s + 2Z_0 Z_s + Z_0^2}, \qquad (6b)$$

calculated using Eqs. (1) and (4) and then making appropriate simplifications for $|Z_a/Z_s| \ll 1$. Figure 2 depicts the waveguide tonehole two-port scattering junction in terms of these reflectances and transmittances. This structure is analogous to the four-multiply Kelly-Lochbaum scattering junction (Kelly and Lochbaum, 1962).

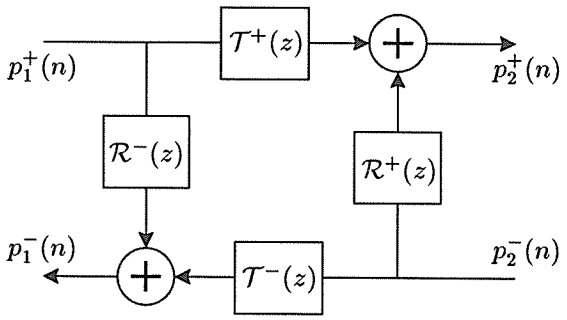

Fig. 2: Digital waveguide tonehole two-port scattering junction.

For the implementation of the reflectances and transmittances given by Eqs. (6a) – (6b) in the digital waveguide structure of Fig. 2, it is necessary to convert the continuous-time filter responses to appropriate discrete-time representations. In this study, use is made of an equation-error minimization technique (Smith, 1983) which matches both frequency response magnitude and phase. This technique is implemented in *MATLAB*® by the function *invfreqz*. Figure 3 plots the responses of second-order discrete-time filters designed to approximate the continuous-time magnitude and phase characteristics of the reflectances for closed and open toneholes. The open-hole discrete-time filter was designed using Kopec's method (Smith, 1983, p. 46), in conjunction with the equation-error method. That is, a one-pole model $\hat{H}_1(z)$ was first fit to the continuous-time response, $H(e^{j\Omega})$. Subsequently, the inverse error spectrum, $\hat{H}_1(e^{j\Omega})/H(e^{j\Omega})$ was modeled with a two-pole digital filter, $\hat{H}_2(z)$. The discrete-time approximation to $H(e^{j\Omega})$ was then given by $\hat{H}_1(z)/\hat{H}_2(z)$. The first step of this design process captures the peaks of the spectral envelope, while the second step models the "dips" in the spectrum. These particular calculations were performed for a tonehole of radius $b = 4.765$ mm, minimum tonehole height $t_w = 3.4$ mm, tonehole radius of curvature $r_c = 0.5$ mm, and air column radius $a = 9.45$ mm. The results of Keefe (1981) were experimentally calibrated for frequencies less than about 5 kHz, so that the continuous-time responses evident in the figures are purely theoretical above this limit. Therefore, the discrete-time filter design process was weighted to produce better matching at low frequencies.

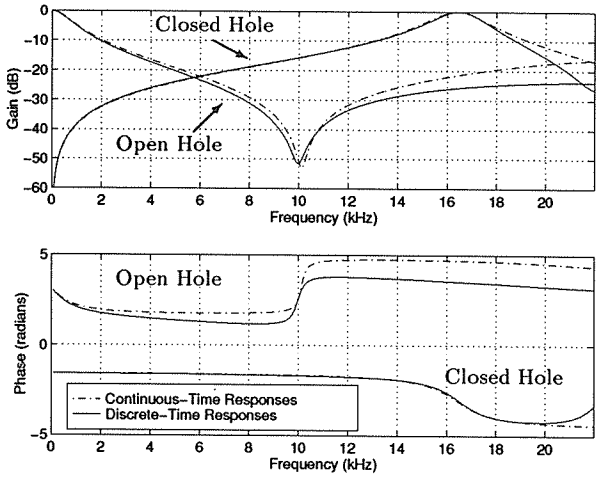

Fig. 3: Two-port tonehole junction closed-hole and open-hole reflectances, derived from Keefe (1981) shunt and series impedance parameters. (top) Reflectance magnitude; (bottom) Reflectance phase.

Figure 4 plots the reflection function calculated for a six-hole flute bore, as described in (Keefe, 1990). The upper plot was calculated using Keefe's frequency-domain transmission matrices, such that the reflection function was determined as the inverse Fourier transform of the corresponding reflection coefficient. This response is equivalent to that provided by Keefe (1990), though scale factor discrepancies exist due to differences in open-end reflection models and lowpass filter responses. The lower plot was calculated from a digital waveguide model using two-port tonehole scattering junctions. Differences between the continuous- and discrete-time results are

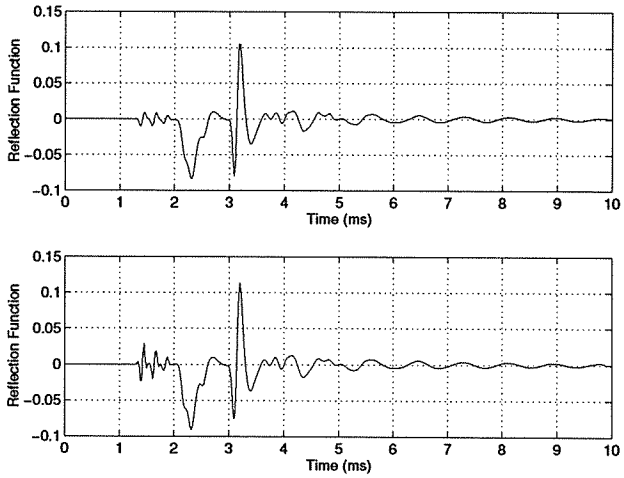

Fig. 4: Reflection functions for note G (three finger holes closed, three finger holes open) on a simple flute [see (Keefe, 1990)]. (top) Transmission-line calculation; (bottom) Digital waveguide two-port tonehole implementation.

most apparent in early, high-frequency, closed-hole reflections. The continuous-time reflection function was low-pass filtered to remove time-domain aliasing effects incurred by the inverse Fourier transform operation and to better correspond with the plots of (Keefe, 1990). By trial and error, a lowpass filter with a cutoff frequency around 4 kHz was found to produce the best match to Keefe's results. The digital waveguide result was obtained at a sampling rate of 44.1 kHz and then lowpass filtered to a 10 kHz bandwidth, corresponding to that of (Keefe, 1990). Further lowpass filtering is inherent from the first-order Lagrangian, delay-line length interpolation technique used in this model (Välimäki, 1995). Because such filtering is applied at different locations along the "bore," a cumulative effect is difficult to accurately determine. The first tonehole reflection is affected by only two interpolation filters, while the second tonehole reflection is affected by four of these filtering operations. This effect is most responsible for the minor discrepancies apparent in the plots.

2 Three-Port Tonehole Model

A tonehole junction may also be represented in the digital waveguide context by a lossless three-port junction. The three-port junction models sound wave interaction at the intersection of the air column and tonehole, as determined by conservation of volume flow and continuity of pressure. Wave propagation within the tonehole itself can subsequently be modeled by another waveguide and the reflection/transmission characteristics at its end by an appropriate digital filter. This tonehole model is then attached to the appropriate branch of the three-port junction. The bore characteristic admittance Y_0 is equal on either side of the junction, while the real tonehole characteristic admittance is Y_{0th}.

The three-port scattering junction equations for pressure traveling-wave components can be determined as

$$\begin{aligned} p_a^-(t) &= r_0 p_a^+(t) + [1+r_0]\, p_b^-(t) \\ &\qquad -2r_0 p_{th}^-(t) \quad (7a) \\ p_b^+(t) &= [1+r_0]\, p_a^+(t) + r_0 p_b^-(t) \\ &\qquad -2r_0 p_{th}^-(t) \quad (7b) \\ p_{th}^+(t) &= [1+r_0]\, p_a^+(t) + [1+r_0]\, p_b^-(t) \\ &\qquad -[1+2r_0]\, p_{th}^-(t), \quad (7c) \end{aligned}$$

where

$$r_0 = \frac{-Y_{0th}}{Y_{0th}+2Y_0} = \frac{-Z_0}{Z_0+2Z_{0th}}. \quad (8)$$

A one-multiply form of the three-port scattering equations is given by

$$\begin{aligned} p_a^-(t) &= p_b^-(t) + w \quad (9a) \\ p_b^+(t) &= p_a^+(t) + w \quad (9b) \\ p_{th}^+(t) &= p_a^+(t) + p_b^-(t) - p_{th}^-(t) + w, \quad (9c) \end{aligned}$$

where

$$w = r_0 \left[p_a^+(t) + p_b^-(t) - 2p_{th}^-(t)\right]. \quad (10)$$

An implementation of these equations in shown in Fig. 5.

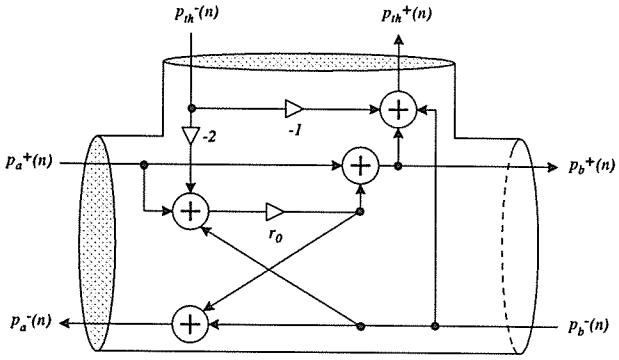

Fig. 5: Tonehole three-port scattering junction implementation in one-multiply form.

To complete the digital waveguide three-port tonehole implementation, it is necessary to determine an appropriate model for the tonehole section itself, and then attach this model to the junction. It is

possible to implement the tonehole structure as a short, fractional delay, digital waveguide and apply an appropriate reflectance at its end. Depending on the tonehole geometry, the reflectance at the end of an open tonehole may be determined from either a flanged or unflanged (Levine and Schwinger, 1948) pipe approximation. The far end of a closed tonehole is appropriately modeled by an infinite impedance (or a pressure reflection without inversion). Given typical tonehole heights, however, a lumped reflectance model of the tonehole, which accounts for both the propagation delay and end reflection is more appropriate and easily implemented with a single low-order digital filter. In this sense, incoming tonehole pressure $p_{th}^-(t)$ is calculated from the outgoing tonehole pressure $p_{th}^+(t)$ and the lumped tonehole driving point reflectance, while the corresponding pressure radiated from the open tonehole is given by convolution of $p_{th}^+(t)$ with the lumped tonehole section transmittance. Figure 6 plots the reflection function obtained for the six-hole flute bore implemented using digital waveguide three-port tonehole junctions. The lumped open-hole reflectance incorporates an unflanged characteristic, while the closed-hole reflectance which best matches the Keefe (1990) data includes no propagation delay within the side branch. Alternatively, the lumped tonehole reflectance filters can be designed from the shunt impedance parameters of Eqs. (3a) and (3b), thus taking advantage of the data of Keefe (1981). The digital waveguide three-port tonehole junction implementation presented here corresponds to the two-port model when series impedance terms are neglected. In general, the series impedance terms are much less critical to the model performance than the shunt impedance, which is demonstrated by the similarity of the results for both implementations. Further, the series terms have more influence on closed-hole results than those for open holes (Keefe, 1981).

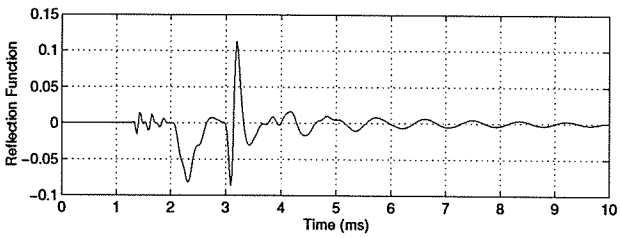

Fig. 6: Reflection function for note G (three finger holes closed, three finger holes open) on a simple flute [see (Keefe, 1990)], determined using a digital waveguide three-port junction tonehole implementation.

3 Conclusions

Current theoretical models of woodwind finger holes can be accurately implemented in the digital waveguide domain. The two-port tonehole waveguide implementation requires four second-order filtering operations per tonehole (details regarding a one-filter form are to be published in the proceedings of the 1997 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics). The three-port implementation requires one multiply and one filtering operation. The results for both implementations are very similar, despite the fact that the three-port model neglects the series impedance terms. A more complete and detailed analysis of this topic, unconstrained by page number limitations, can be found in (Scavone, 1997).

References

Keefe, D. H. (1981). *Woodwind Tone-hole Acoustics and the Spectrum Transformation Function.* Ph.D. thesis, Case Western Reserve University.

Keefe, D. H. (1990). Woodwind air column models. *J. Acoust. Soc. Am.*, 88(1):35–51.

Kelly, Jr., J. L. and Lochbaum, C. C. (1962). Speech synthesis. In *Proc. Fourth Int. Congress on Acoustics*, pp. 1–4, Copenhagen, Denmark. Paper G42.

Levine, H. and Schwinger, J. (1948). On the radiation of sound from an unflanged circular pipe. *Phys. Rev.*, 73(4):383–406.

Scavone, G. P. (1997). *An Acoustic Anaylsis of Single-Reed Woodwind Instruments with an Emphasis on Design and Performance Issues and Digital Waveguide Modeling Techniques.* Ph.D. thesis, Music Dept., Stanford University. Available as CCRMA Technical Report No. STAN-M-100 or from ftp://ccrma-ftp.stanford.edu/pub/Publications/Theses/Gary-ScavoneThesis/.

Smith, J. O. (1983). *Techniques for Digital Filter Design and System Identification with Application to the Violin.* Ph.D. thesis, Elec. Eng. Dept., Stanford University.

Välimäki, V. (1995). *Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters.* Ph.D. thesis, Helsinki University of Technology, Faculty of Electrical Engineering, Laboratory of Acoustic and Audio Signal Processing, Espoo, Finland, Report no. 37.

Nonlinear Commuted Synthesis of Bowed Strings*

Julius O. Smith III
CCRMA, Music Department, Stanford University
jos@ccrma.stanford.edu, http://www-ccrma.stanford.edu/~jos

Abstract

A commuted-synthesis model for bowed strings is driven by a separate nonlinear model of bowed-string dynamics. This gives the desirable combination of a full range of complex bow-string interaction behavior together with an efficiently implemented body resonator. A "single-hair bow" may control a pulsed-noise version which provide the effects of multiple bow hairs. The pulsed noise may also include qualitatively the impulse responses of commuted high-frequency body modes.

1 Introduction

According to prevalent theories of bow-string interaction [McIntyre and Woodhouse 1979, Guettler 1992], disturbances sent out by the stick-slip process along the string are fundamentally *impulsive* in nature. That is, the bow is normally either sticking or slipping against the string, and the main excitation events on the string occur when the slipping starts or ends, at which point there is a narrow acceleration pulse sent out in both directions along the string. (There is also sliding noise during slipping each period, but that can be dealt with separately.) Both the Helmholtz [1863] and Raman [1918] models of bowed string behavior consist only of sparse acceleration impulses on the string. Raman's theory, in fact, classifies the various motions according to how many impulses there are per period. Basic Helmholtz motion only consists of one impulse per period, while other modes, such as "surface sounds" generated by "multiple slips," or "multiple flybacks," consist of two or more acceleration impulses per period.

The implication of any "sparse impulse model" of bowed-string interaction is that it can be used to efficiently drive a commuted synthesis implementation for bowed strings [Smith 1993, Jaffe and Smith 1995]. The advantage of commuted synthesis is that a potentially enormous recursive digital filter representing the resonating body is avoided. When an impulse reaches the bridge, a body impulse response (BIR) is "triggered" at the amplitude of the impulse. The commuted synthesis implementation thus "watches" impulses arriving at the bridge in the bowed-string model, and instantiates a BIR playback into a separate string model on the arrival of each impulse. (BIR playbacks which overlap in time are summed.) A BIR playback may be implemented, for example, using a wavetable oscillator in "one-shot" mode. The variable playback rate normally available in such an oscillator can be used to modulate apparent "body size" [Cook 1996, Mandolin.cpp]. The impulse-triggered BIR playback scheme can be classified as an efficient "sparse-input FIR filter" implementation of the body resonator. For simple Helmholtz motion, this model reduces to the original bowed-string commuted-synthesis model, except that we may now generate automatically impulse amplitude and timing information from the bow-string interaction model, and we can use physical bow force, position, and velocity signals as the control inputs. In this way, we obtain the reduced computational cost of commuted synthesis, at least during smooth playing, while allowing for fully general interaction between the bow and string.

2 Nonlinear Commuted Model

The basic idea of commuted synthesis is to interchange the order of implementation of the string and the body resonator, as depicted in Fig. 1.

The bowed string synthesizer of the present paper is shown in Fig. 2. The bottom half is Fig. 1c, with an external trigger input, and some further details regarding pulsed noise generation. The top half of Fig. 2 provides an explicit model of bow-string dynamics. The "Impulse Prioritizer" measures the timing and amplitude of the largest impulses in the string waveform at the bridge and passes on the most important

*Expanded version for CCRMA affiliates. A shorter four-page version was submitted to the 1997 International Computer Music Conference.

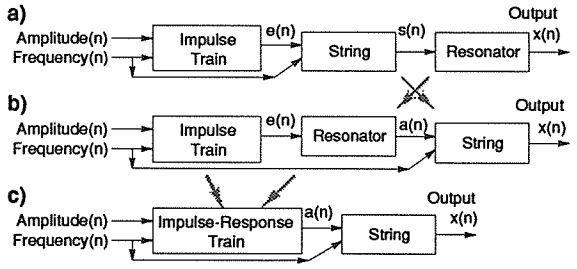

Figure 1: a) Simplified bowed string model, including only amplitude, pitch, and vibrato control capability. b) Equivalent diagram with resonator and string commuted. c) Equivalent diagram in which the resonator impulse response is played into the string each pitch period.

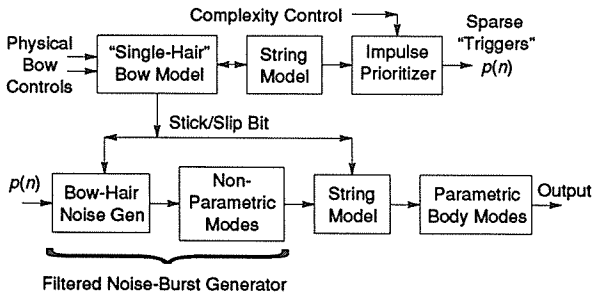

Figure 2: Commuted bowed string synthesis model driven by a separate bow-string model exhibiting full nonlinear dynamic behavior.

ones subject to complexity constraints. The second string which is driven by the BIR oscillators may be a digital waveguide model driven at the bowing point, or it may consist of an equivalent feedforward comb filter followed by a filtered delay loop. However, the advantage of a a full waveguide model of the string [Smith 1986] is that the time-varying, nonlinear, partial termination of the string by the bow can be more conveniently implemented.

The Stick/Slip Bit can be used to switch between two models of partial string termination by the bow. For more accurate control of string damping by the bow, the contact force, relative velocity, position along the string, and bow angle can all be used to determine the frequency-dependent scattering junction created by the bow on the string [Smith 1986]. It was found empirically that significant damping of the string by the bow is necessary for obtaining robust Helmholtz motion; otherwise, excessive ringing of the string segment between the bow and nut tends to cause slipping at times disruptive to the Helmholtz motion. Intuitively, one of the two "Helmholtz corners" sent in opposite directions along the string on each slip/stick impulse must be "filtered out" by the bow, while the other is "amplified" by the stick/slip process. Graphical animation of the bowed string motion was found to be very helpful for determining qualitative factors such as this.

3 Nonlinear Bow Friction

For this study, a simplified bow-string interaction model was implemented having the following characteristics:

- Static frictional "release force" is a multiple of the vertical bow force.
- Dynamic frictional force is small and fixed (independent of bow force).
- The "capture force" is somewhat smaller than the "release force" but also a multiple of the vertical bow force.
- One bit of state is maintained ("sticking" vs. "slipping") in order to distinguish whether to use capture or release maximum forces.

The use of different maximum forces for release versus capture is motivated by the recent findings that there is evidence that the bow rosin *melts* during slipping and refreezes during sticking [Smith 1990].

4 Friction Impulse Detection

The output of the bow-string simulation must be converted to discrete trigger events, with each trigger initiating playback of the body impulse response (BIR). Ideally, we would like a means of "thinning" the impulses coming from the bridge so as to keep the most important ones and neglect the least important ones to the degree necessary to meet computational resource restrictions.

There are several alternative impulse thinning schemes. Perhaps the simplest is to set an impulse amplitude *threshold*, such as ten percent of the expected main impulse amplitude, such that any impulse over the threshold in magnitude is passed on as a trigger, and anything smaller is suppressed. When the threshold is crossed by the absolute value of the bridge acceleration waveform in an upward direction, the next local maximum is taken to determine the impulse amplitude and timing. No further impulses

are accepted until the bridge acceleration falls below the threshold. As a further refinement, the samples on either side of the local maximum can be used to quadratically interpolate the peak, as is typically done for spectral peaks; alternatively, or in addition, the bow-string simulation can be run at a higher sampling rate than the commuted synthesis unit in order to further improve the impulse timing accuracy.

The simple threshold method does not introduce latency, which is important in the real-time case, but it does not enable optimal impulse detection methods and there is no direct control over complexity (it is not easily known in advance what threshold will thin the impulse stream to the necessary extent). An indirect control over complexity is obtained by setting the threshold dynamically as a function of the number of overlapping BIRs. In this way, the threshold can be lifted to increase the thinning when the complexity becomes too great. An advantage of this thinning algorithm is that it doesn't matter what the source of complexity is. For example, impulses may be thinned because the pitch went higher causing more BIR overlap, or because other voices came in reducing the available number of BIR oscillators, or because the end user changed a preference specifying an upper limit on computing resources to be devoted to sound synthesis on a general purpose computer.

A more direct impulse thinning scheme which introduces one period of latency delay is as follows: The most recent period of the bridge signal is kept in a circular buffer at all times. Let N_e denote the maximum number of stick-slip events allowed per period P. To restrict behavior to basic Helmholtz motion, N_e can be set to 1. To allow second-order Raman motion, $N_e = 2$ would be appropriate, and so forth. At each time step, the largest N_e peaks in the last period are defined as the impulses to send out. Since there is one period of latency, it is always the case that the emitted impulses are the most important ones within the past period. Having a period of "look ahead" enables use of more sophisticated peak detection schemes than the simple local-maximum-after-threshold-crossing method.

A variation on the threshold method which does not need threshold adaption for complexity control is analogous to *voice allocation* in polyphonic synthesizers: When an impulse crosses a nominal threshold level, the next local maximum triggers a BIR playback unless (1) all playback units are busy *and* (2) the desired playback amplitude is *smaller* than that of *all* of the playing BIRs. When all BIR units are busy but one of them is deemed less important than the desired new BIR, the least important BIR is *preempted*, interrupting its playback and restarting it at the desired amplitude for the new BIR playback.

5 Pulsed Noise

A stick-slip event never involves only one bow hair, and during the slipping interval, or string "flyback," there is a soft noise burst which is audible, especially at close range. It is well known that pulsed noise is an important feature of high quality bowed-string synthesis as well as other instruments [Chafe 1990]. The Stick/Slip Bit provided by the bow-string contact model (see Fig. 2) indicates when sliding noise is appropriate. As in the case of the time-varying string-damping discussed above, more refined noise-generation models can be devised based on the bow force, differential velocity, and position information available from the bow-string simulator, as well as an external "bow angle" control.

When the resonating body transfer function is *factored* [Karjalainen and Smith 1996] into slowly decaying modes (implemented parametrically using recursive filters and not necessarily commuted) and rapidly decaying modes (which are commuted and used in nonparametric form as impulse response data), the commuted nonparametric impulse response is qualitatively a short, *high-frequency noise burst*, since it consists of the impulse responses of thousands of high-frequency, highly damped modes. In principle, this "damped-modes-noise-burst" should be convolved with the noise arising from the slipping bow. In other words, the string excitation for each stick-slip event can be modeled as a filtered noise burst which includes both the highly damped resonator modes and the bow noise.

6 Simulation Results

Figure 3 displays waveforms generated by the bow-string model given a constant bow force, velocity, and position. The frictional force applied to the string by the bow can be seen to diminish as the oscillation develops. The string displacement near the bridge clearly exhibits the single main impulse once per period associated with canonical Helmholtz bowed-string motion; there are also many secondary impulses associated with the ringing of the piece of the string between the bridge and the bow. The complexity control will determine whether these secondary impulses are included or suppressed.

Figure 4 illustrates the samples of bridge displacement waveform over a longer period of time. Note that each main Helmholtz impulse plots as two

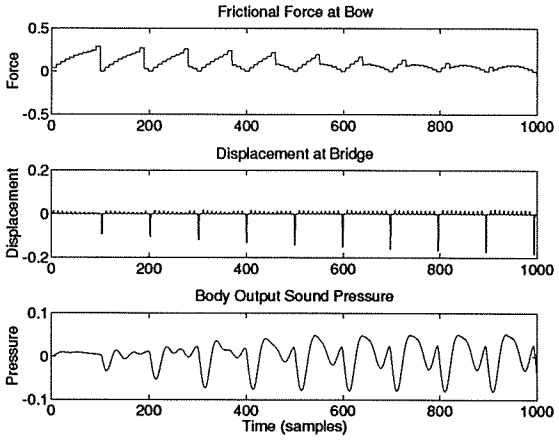

Figure 3: Output of the bow-string model before extracting bridge imimpulses. Top: Frictional force between bow and string. Middle: String displacement 1/2 sample from the bridge. Bottom: Sound pressure radiated from simulated body filter. Bowing parameters (fixed): speed 15 cm/sec, force 20 grams, position 3 cm from bridge. A two-pole, two-zero bridge-filter for a digital waveguide string model was calibrated to measurements of violin pizzicato waveforms. A torsional-wave loss coefficient of 0.9 was implemented at the bow at all times.

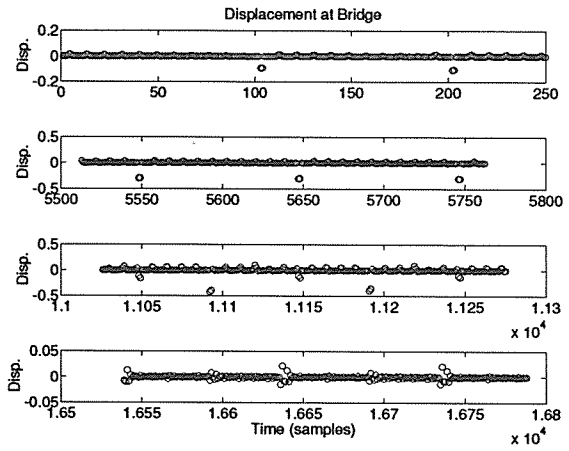

Figure 4: String displacement 1/2 sample from the bridge over four short time intervals spanning 1.7 seconds.

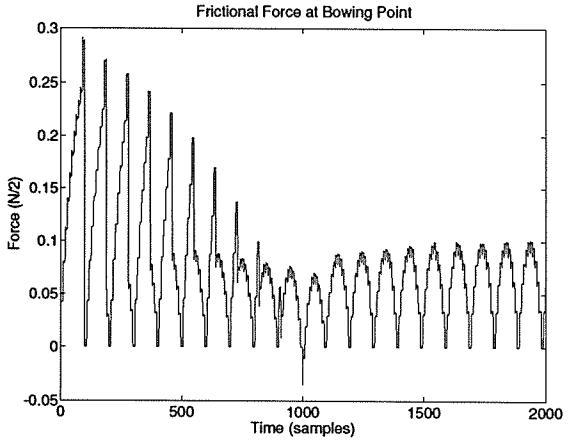

Figure 5: Close up of the frictional force waveform during the initial attack.

adjacent samples, indicating that a single-sample impulse is traveling on the string. (The observation point is 1/2 spatial sample from the bridge, so that a single impulse at the bridge appears twice, both before and after reflection at the bridge.) Note also that late in the stroke, a strong secondary impulse has developed, making the sound tend toward an octave higher. This "sul ponticello" sound is associated with insufficient bow force.

Figure 5 gives a close-up of the frictional force during the initial attack transient. As can be seen, even though the applied bow force and velocity are constant, a highly complex interaction occurs between the bow and string.

Figure 6 shows an overlay of the first 40 periods of oscillation of the bowed string, with each string snapshot taken slightly later than one period after the previous, and the first snapshot being taken at time zero. The bow is at the sharp upper corner on the left. Note that the vertical scale is highly magnified relative to the horizontal scale. There is also some distortion in the string shape resulting from the lumping of the string losses at the bridge and bowing point, as is typical in waveguide string modeling.

7 Conclusions

The commuted bowed-string synthesis model was extended to incorporate driving information from a nonlinear model of bowed-string dynamics. The formulation allows a simplified "single-hair bow" to control a pulsed-noise driven commuted synthesis model, thereby simulating a full-width bow in the final sound quality. Commuting only the fastest decaying (high frequency) body modes results in a short, damped impulse response which can be regarded as a component of the pulsed noise. In summary, driving a commuted-synthesis model for bowed strings from a nonlinear model of bowed-string dynamics gives the desirable combination of a full range of complex bow-string interaction behavior together with an reduce-

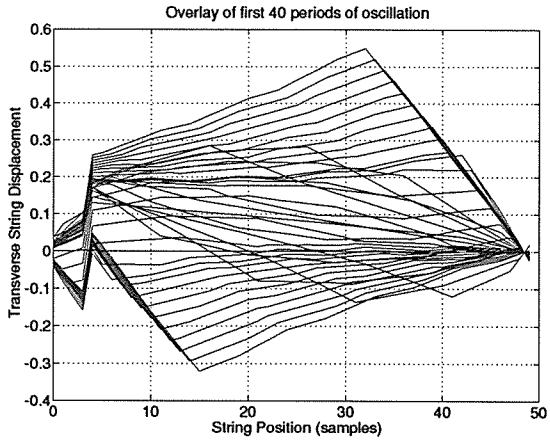

Figure 6: Snapshots of string state for first 40 periods of oscillation.

complexity body resonator.

References

[Chafe 1990] Chafe, C. 1990. "Pulsed Noise in Self-Sustained Oscillations of Musical Instruments." *In: Proc. Int. Conf. Acoustics, Speech, and Signal Processing, Albuquerque.* New York: IEEE Press. Available as CCRMA Technical Report STAN-M-65, Music Dept., Stanford University.

[Cook 1996] Cook, P. R. 1996. "Synthesis ToolKit in C++, Version 1.0." *In: SIGGRAPH Proceedings.* Assoc. Comp. Mach. See http://www.cs.princeton.edu/~prc/NewWork.html for a copy of this paper and the software. (All simulations for this paper were done using the STK and Matlab.)

[Guettler 1992] Guettler, K. 1992. "The Bowed String Computer Simulated — Some Characteristic Features of the Attack." *Catgut Acoustical Soc. Journal*, 2(2):22–26. Series II.

[Jaffe and Smith 1995] Jaffe, D. A., and J. O. Smith. 1995. "Performance Expression in Commuted Waveguide Synthesis of Bowed Strings." *Pages 343–346 of: Proc. 1995 Int. Computer Music Conf., Banff.* Computer Music Association (CMA).

[Karjalainen and Smith 1996] Karjalainen, M., and J. O. Smith. 1996. "Body Modeling Techniques for String Instrument Synthesis." *In: Proc. 1996 Int. Computer Music Conf., Hong Kong.* CMA.

[McIntyre and Woodhouse 1979] McIntyre, M. E., and J. Woodhouse. 1979. "On the Fundamentals of Bowed String Dynamics." *Acustica*, 43(2):93–108.

[Pickering 1991] Pickering, N. C. 1991. *The Bowed String.* Mattituck NY: Amereon, Ltd. Also available from Bowed Instruments, 23 Culver Hill, Southampton, NY 11968.

[Pitteroff 1993] Pitteroff, R. 1993. "Modelling of the bowed string taking into account the width of the bow." *Pages 407–410 of: Proc. Stockholm Musical Acoustics Conference (SMAC-93).* Stockholm: Royal Swedish Academy of Music.

[Raman 1918] Raman, C. V. 1918. "On the Mechanical Theory of Vibrations of Bowed Strings, etc." *Indian Assoc. Cult. Sci. Bull.*, 15:1–158.

[Smith 1990] Smith, J. H. 1990. *Stick-Slip Vibration and its Constitutive Laws.* Ph.D. thesis, Cambridge Univ.

[Smith 1986] Smith, J. O. 1986. "Efficient Simulation of the Reed-Bore and Bow-String Mechanisms." *Pages 275–280 of: Proc. 1986 Int. Computer Music Conf., The Hague.* CMA.

[Smith 1993] Smith, J. O. 1993. "Efficient Synthesis of Stringed Musical Instruments." *Pages 64–71 of: Proc. 1993 Int. Computer Music Conf., Tokyo.* CMA.

[Smith 1996] Smith, J. O. 1996. "Physical Modeling Synthesis Update." *Computer Music J.*, 20(2):44–56. Available online at http://www-ccrma.stanford.edu/~jos/.

[von Helmholtz 1863] von Helmholtz, H. L. F. 1863. *On the Sensations of Tone as a Physiological Basis for the Theory of Music.* New York: Dover. English translation by A. J. Ellis. 1954.

8 Appendix: Selected Software Items

All simulations for this paper were carried out using Perry Cook's Synthesis ToolKit (STK) in C++ [Cook 1996]. The method `stringVelocityAtPosition(int position)` below can be added to bowed.cpp in the STK to facilitate extracting the string state for display as shown in Fig. 6. (Animations of bowed-string motion using this added method were found to be especially valuable for obtaining insight into bowed string dynamics.)

```
MY_FLOAT BowedStr :: stringVelocityAtPosition(int p) /* p from 0 to nsamples-1 */
{
    int bdelrt = (int)bridgeDelay->delay()+1; /* bow-to-bridge-to-bow + p.l. delay */
    int ndelrt = (int)neckDelay->delay()+1; /* bow-to-nut-to-bow + p.l. delay */
    int bdel = bdelrt >> 1; /* number of spatial samples from bridge to bow */
    int ndx = bdel - p; /* convert (0:N-1) position to delay (1:N) on left */
    MY_FLOAT leftGoingAtP;
    MY_FLOAT rightGoingAtP;
    if (p < bdel) {
        /* "Now" is always where the InPoint points which is not yet written */
        /* OutPoint points to "now - delay" */
        /* Bridge is on the left, nut on the right */
        /* "Now" is at the bow */
        /* Position zero is at far left = half way along delay line */
        leftGoingAtP = bridgeDelay->contentsAtNowMinus(ndx);
        int rndx = bdelrt-ndx+1;
        if (rndx < bdelrt) { /* last sample delay resides in lastOutput variable */
            rightGoingAtP = -bridgeDelay->contentsAtNowMinus(rndx);
        } else {
            rightGoingAtP = -bridgeDelay->lastOut();
        }
    } else { /* nut side */
        ndx = p - bdel + 1; /* convert (0:N-1) position to delay (1:N) */
        rightGoingAtP = neckDelay->contentsAtNowMinus(ndx);
        int lndx = ndelrt-ndx+1;
        if (lndx < ndelrt) {
            leftGoingAtP = -neckDelay->contentsAtNowMinus(lndx);
        } else {
            leftGoingAtP = -neckDelay->lastOut();
        }
    }
    return rightGoingAtP + leftGoingAtP;
}
```

Usage of the above method is illustrated in the following code fragment:

```
MY_FLOAT stringState[MAXPERIOD];
for (i=0;i<period/2;i++)
    stringState[i] = 0;
for (i=0;i<samples;i++) { /* main sample loop */
    ...
    if (i<20*period) {
        MY_FLOAT v = 0.0;
        long len = period/2;
        for (int j=0; j<len; j++) {
            v = vscale*vln->stringVelocityAtPosition(j); /* m/s */
            stringState[j] += v*ONE_OVER_SRATE; /* m */
            stringOut->tick(STRINGSCALING*stringState[j]); /* to soundfile */
        }
    }
```

The following matlab function was used to generate highly helpful animations of the string state. Each string snapshot was written successively into one long sound file, and this routine was called with the sound data along with M set to the snapshot length in samples:

```
function out=datamovie(in,M,sleep,ax);
%DATAMOVIE   datamovie(in,M,sleep,ax);
%           Display sequence of data frames of length M.
%           If sleep>0, that many cycles are waited between plots.
%           If sleep == -1, RETURN is needed to advance to the next plot.
%           If ax is a quoted string, "axis(ax)" is called.
clf;
if (nargin<2), M=length(in); end
if (nargin<3), sleep=0; end
if (nargin<4), ax = [1 M min(in) 1.1*max(in)]; end
skip = M; h=plot(in(1:M),'erasemode','background'); axis(ax); drawnow;
if sleep == -1, disp '*** PAUSING *** RETURN to continue'; pause; end
for i=1:(length(in)-M)/skip
    set(h,'ydata',in(skip*i+1:skip*i+M));
    if sleep == -1, disp '*** PAUSING *** RETURN to continue'; pause;
    elseif sleep>0, for j=1:sleep, y = tan(j); end; end
end
```

Usage of the datamovie function is illustrated by the matlab script below:

```
% seestr.m - matlab script for viewing bowed string waveshape evolution
ilen = 50;              % Number of spatial samples along string
sleep = 0;              % pause/speed control to datamovie
name1 = 'string'; [strdata fs len header] = loadsig(name1); % for NeXT .snd files
strdata = strdata/32768.0;
datamovie(strdata,ilen,sleep);
```

Applying Root-Locus Techniques to the Analysis of Coupled Modes in Piano Strings

Timothy S. Stilson
Center for Computer Research in Music and Acoustics, Stanford University
stilti@ccrma.stanford.edu, http://www-ccrma.stanford.edu/~stilti

Abstract

Previous work in the study of coupled piano string behavior has focused analytically on the interaction of a pair of coupled modes, noting that the rest of the string modes couple similarly. In this paper, multi-string coupling is analyzed with the root-locus method, which describes the movement of system poles under variations in a single parameter, such as coupling magnitude. Many effects, for example, two-stage decay, can be understood in terms of system pole location. Often, an intuitive understanding of pole movement under variations in a parameter, such as a single mode frequency or coupling coefficient, can be developed. Three-mode coupling is explored and interpreted, as is variation in coupling behavior at different string harmonics.

1 Introduction

Previous work in the study of coupled piano string behavior ([1], [2], [3]) has either focused analytically on the interaction of a pair of coupled modes, noting that the rest of the string modes couple similarly, or has studied the coupling experimentally. The equations describing three-string coupling (along the lines of [3]) become extremely complex and one can easily become lost in interpreting them. In this paper, we will show how one can analyze the coupling behavior in terms of the *Root Locus* analysis method, which analyzes the location of the poles of a closed-loop linear system according to the pole and zeros of the open-loop system and under variation of the feedback gain. Root Loci were for decades drawn by hand, so that a lore was developed about patterns that appear in root loci; this lore represents an intuition that can be acquired and applied to get a feel for the behaviors of closed-loop systems — coupled strings, in this case.

The coupling of two and three modes, as occurs in sets of unison piano strings, is studied, along with the coupling behavior of multiple groups of modes, such as the multiple harmonics of the strings. Many effects, for example, beating and two-stage decay, can be understood in terms of the system pole locations (since for impulsively driven systems such as pianos, the normal behavior of the system is given by the impulse response of the system), which makes the root-loci directly interpretable. Furthermore, rather complex, frequency-dependent coupling impedances can easily be included in the analysis, allowing an exploration of the variation in coupling behavior at different string harmonics.

2 Root Locus Analysis

This technique, which originated in the analysis of linear feedback control systems, analyzed the poles of a closed-loop feedback system in terms of the open-loop transfer function and the (variable) loop gain. Textbooks on classical control systems, such as [4], provide rules on drawing root-loci. The most important rule to note is that the closed-loop poles coincide with the open-loop poles when $k = 0$, and move to coincide with the open-loop zeros as $k \to \infty$. Thus we gather quite a bit of information simply by plotting the open loop poles and zeros. Another rule is that the paths of the poles are given by the equation $\angle(GH) = \pi$. We can use this rule to draw root-loci of non-rational linear systems, which show up in the analysis of continuous-time string coupling.

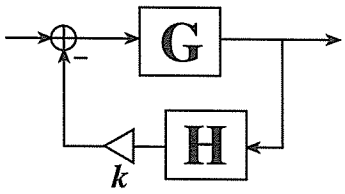

Figure 1: Linear system in Root-Locus Form

3 Coupled Modes

Let $G_{\text{Forward}} = \frac{1}{s-p_1} + \frac{1}{s-p_2}$, and let $G_{load} = e^{j\phi}$. We can then interpret k as the magnitude of the coupling between the two open-loop modes p_1 and p_2.

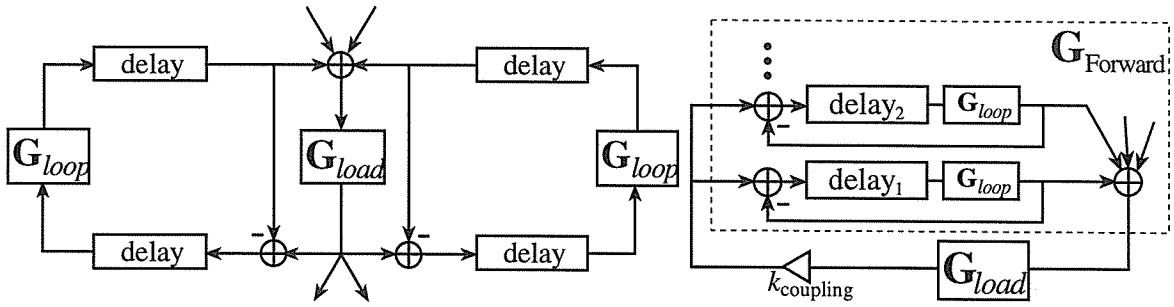

Figure 2: Waveguide representation of N coupled strings, and rearranged into Root-Locus form

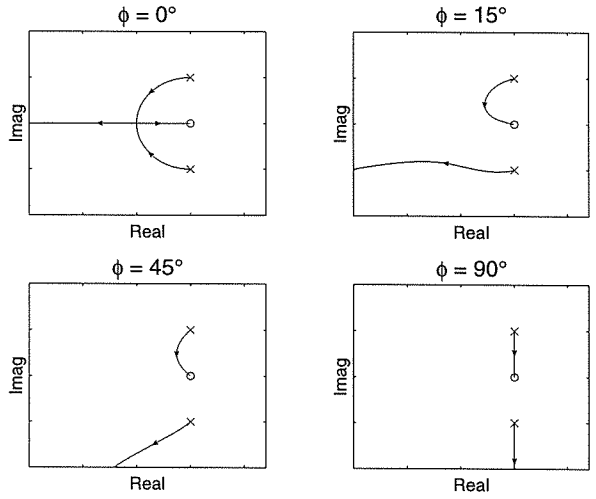

Figure 3: Sample 2-Coupled-Mode Loci

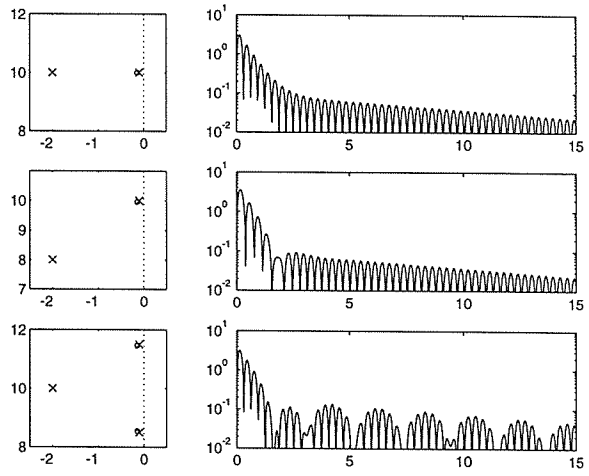

Figure 4: Rectified impulse responses for various two-stage closed-loop pole configurations

We draw root-loci for $k > 0$, and for different values of ϕ (Figure 3). For small coupling (small k), the poles stay near the open-loop poles. The impulse response of the system in this case would have betaing between the modes. As the coupling gets stronger, the poles move into a region where one pole has a faster decay than the other, giving the well-known two-stage decay. In this region, the effect of the coupling angle is to "rotate" the root locus, which detunes the modes when they are in the two-stage region. This detuning accounts for the momentary dip of the decay at the crossover between stages ([1], Figure 11, or [3], Figure 6b). The rotation of the locus also explains the "mode repulsion" effect for reactive impedances (lower-right plot: the 90-degree coupling case).

It is important to note that the *shapes* of the loci are relatively independent of the actual mistuning of the modes. These loci will look the same for any vertical spacing of the two modes. The distance between the open-loop poles *does* affect the range of k over which different behaviors will occur: the closed loop poles land at the intersections of the $\angle(GH) = \pi$ and $|GH| = 1/k$ contours; when the poles are closer together, the magnitude of GH is larger in the region where the root-locus has interesting behavior (like where the locus tracks intersect in the real-coupling case). The main result of this is that one can also move the system between beating and two-stage decay by fixing k and adjusting the mistuning of the modes (which is how Weinreich did it in [1]).

4 Coupled Strings

Referencing Figure 2, we can set up the root-locus for a system of coupled strings by letting:

$$G_{\text{Forward}} = \sum_{i=1}^{N} \frac{-G_{loop_i} e^{-sT_i}}{1 - G_{loop_i} e^{-sT_i}} \qquad (1)$$

Where the T_i are the round-trip delays of each string, and G_{loop_i} are the lumped round-trip loss of the strings. We can draw the root locus vs. k_{coupling} of the coupled system by evaluating the contour $\angle\left(G_{\text{Forward}}(s)\, G_{load}(s)\right) = 0$ in the s-plane.[1] We can determine the closed-loop pole

[1] We evaluate at $\angle(\bullet) = 0$ instead of $\angle(\bullet) = \pi$ because the system is missing the extra sign inversion in the loop that the system in Figure 1 has due to the subtraction in the loop.

locations for a given k_{coupling} by evaluating the contour $|G_{\text{Forward}}(s)\,G_{load}(s)| = 1/k_{\text{coupling}}$ and locating its intersections with the root locus.

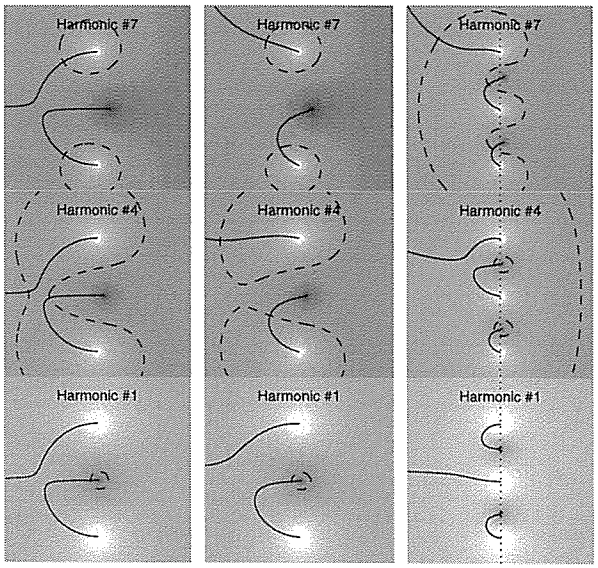

Figure 5: String Loci, *left:* two strings, real coupling, *middle:* two strings, one-pole G_{load}, *right:* three strings, one-pole G_{load}.

In Figure 5, we have plotted select sections of the root loci for three different systems. In the left system, we see three harmonics for the case of two-string coupling with a real scalar G_{load}. This case shows the fact that the distance between the poles will change the 'effective' coupling strength: the low harmonics, being closer, are strongly into two-stage decay, with the fast mode out of the picture to the left; whereas at the higher harmonics, the detuned open-loop poles end up further apart, so that at the seventh harmonic, the poles are still in a beating configuration. This emphasizes that different harmonics will couple differently with the same coupling constant, simply due to their pole spacing.

The middle case has G_{load} set to a onepole lowpass filter $\frac{a}{s+a}$. This coupling is now frequency dependant, so that each harmonic will see a different coupling factor. Since the phase of the onepole filter approaches -90 degrees at high frequencies, we would expect that higher harmonics will display coupling patterns that are more and more rotated, along with the distance-induced reduction in coupling, which can be seen in the figure.

The right-hand case has three strings, along with the one-pole frequency dependent coupling. An important effect can be seen at the first harmonic, where the coupling is nearly real: the sum of the three modes has two zeros "in-between" the poles, this causes *two* of the poles to stay at slow decay when the system is strongly coupled, so that only the third pole goes into fast decay. Thus, during the second stage of decay, the two slow modes will beat with each other, an effect that is seen in [2] and [3].

5 Conclusions

The Root Locus visualizes the process of mode coupling in systems, and can provide intuition on the coupling behavior. Some important facts about multi-string coupling are deduced from the root loci:

- In 3-mode coupling, two modes stay at slow decays, this gives the beating in the second stage
- Different string harmonics couple differently:
  - Higher harmonics are more detuned (absolute detuning, not relative), this makes the higher harmonics "less coupled"
  - Frequency-dependent coupling causes each harmonic to couple differently, for example each harmonic couples with a different coupling phase angle.

References

[1] Weinreich, G. 1977. "Coupled Piano Strings", J. Acoust. Soc. Am. Vol. 62, No. 6, pp. 1474-84.

[2] Hundley, C. et al 1978. "Factors Contributing to the Multiple Rate of Piano Tone Decay", J. Acoust. Soc. Am. Vol. 64, No. 4, pp. 1303-09.

[3] Nakamura, I. 1989. "Fundamental Theory and Computer simulation of the Decay Characteristics of Piano Sound", J. Acoust. Soc. Jpn. Vol. 10, No. 5, pp. 289-97.

[4] Franklin, G., Powell, J.D., Emami-Naeini, A. 1994. *Feedback Control of Dynamic Systems,* New York: Addison-Wesley, pp. 243-336.

[5] Smith, J.O. 1993. "Efficient Synthesis of Stringed Musical Instruments", ICMC93, Tokyo, pp.64-71.

The annotated viewgraphs of a talk on this topic, including a more tutorial discussion and more examples, is available at the web site (see the author line) at `http://.../~stilti/papers/RLTalk2.pdf`

An HTTP Interface to Common Music

Heinrich Taube
School of Music, University of Illinois
taube@uiuc.edu

Tobias Kunze
CCRMA, Stanford University
tkunze@stanford.edu
http://www.stanford.edu/~tkunze

Abstract

This paper describes the addition of web-compatible network support to Common Music for a large subset of its functionality. The core score description language and most output capabilities reside on a serving machine and communicate via a native LISP HTTP server with any authorized machine in the local or global network. Accessing the system via the network protocol is intended to enhance existing functionalty rather than replace local interaction with the software. The choice of HTTP and its related protocols is implied by the goal of providing a composition server. The wide availability of web browsers and other clients as well as their inherent platform-independency and support for executable code (Java) set web-based access methods well ahead of other options. Adding server functionality to the composition software relieves weaker machines from running a full-fledged LISP environment and enables all clients to make use of scarce or non-standard resource on the server side, such as platform-specific target software or sound synthesis capabilities. In addition, a client-server design of composition software facilitates greatly lab administration and classroom teaching. Although moving the system in the direction of a client-server architecture seems desireable from a user's or administrator's point of view, we expect composers to benefit most from the ease with which other software clients will be able to use Common Music's compositional services.

1 Composition Server Model

1.1 Motivation

Common Music is a portable composition environment that is able to control a number of different synthesis languages. The system is implemented in Common LISP and C and runs on most computers available today. Though Common Music is highly portable, its kernel must be customized to particular machine configurations when the software is installed. Until recently the portable nature of the software necessitated a number of compromises in system functionality and support for any given local configuration.

The main purpose of this project is to leverage recent HTTP developments to solve local configuration constraints inherent in the portable nature of Common Music. In some sense, this move to a client-server architecture is a natural outcome of Common Music's central aims of portability and synthesis independence. By shifting some of CM's features to HTTP, a number of important features are gained. In particular, it

- relieves weak machines from constraints imposed by local configurations
- gives access to expensive resources such as synthesis services
- allows audio, MIDI, graphics, etc. to be streamed across the network in realtime
- provides a uniform GUI interface
- facilitates classroom teaching
- facilitates system administration.

1.2 Stand-Alone Configuration Constraints

Limits of the local host affect Common Music in several areas. With respect to implementation, the software's portability depends on a number of different, unequally featured Common LISP implementations. For example, some implementations provide CLOS, some do not; some provide an interface to a native windowing system, others do not; some have native compilers while others byte compile; some provide dynamic foreign function loading and others do not. The net effect of the uneven implementation features is that no single machine supports a "fully-featured"

	GUI	MIDI Driver	MIDI File	CLM	Csound	MusicKit	RT	HTML
Mac OS	•	•	•	•	•			•
Windows '95			•		•			•
SGI IRIX		•	•	•	•		•	•
NeXTStep		•	•	•	•	•	•	•
Linux			•	•	•			•

Table 1: Example CM feature support across five local configurations.

runtime configuration. What is worse, the dependency on LISP vendors prohibits the system from running in some hardware environments that would otherwise be attractive. For example, Windows is an important enviroment from the standpoint of market/cost/performance, but due to the cost/quality of vendor support it is currently not a hospitable porting target for Common Music's kernel (cf. Table 1).

1.3 Accessing Synthesis Services

The host computer also affects the set of synthesis options a locally configured Common Music image will control. Although the system supports many different synthesis languages, a composer is actually limited to those languages that can run on the local CPU. Luckily, there are several good synthesis packages (such as CLM and Csound) that are also highly portable. But the fact remains that there are many more computers that can run compositional algorithms with adequate speed than there are computers suitable to executing synthesis algorithms. Moving to a client/server model means that a composer can work with synthesis languages and hardware resources that are not available on the local host. In this model, a remote server equipped with a fast CPU and adequate memory renders sound and returns audio or MIDI streams, or general control information to the local client's plugins to play. The availablility of audio and MIDI streaming in real-time also means that CM no longer needs to provide some of its scheduling and MIDI driver support currently required for each local port.

1.4 Towards a Uniform GUI

The most severe penalty for making CM a portable system has been its relatively weak support for GUI tools. The primary cause of this is the ANSI Common LISP Standard itself, which does not address GUI issues at all. This means that to provide a "portable" LISP-based graphical interface, a developer must completely reimplement the interface for every machine/LISP combination. For this reason, Common Music's GUI (*Capella*) currently runs only on the Macintosh. But the past few years have witnessed a explosive growth in HTTP GUI development; at this point, HTTP support is more active and broad-based than that provided by Common LISP vendors. While HTTP graphics is still rather primitive, its central emphasis on hypertext-based presentation is consistent with the description of algorithmic processes in Common Music. In fact, system documentation, tutorials, dictionaries and ancillary documentation have already been in HTML format for several years. It is now a natural step to extend this support into the presentation of the system itself.

1.5 Inter-Application Services

Providing a simple, network-oriented public interface to Common Music also enables other software clients to make use of its services and thus to enhance their functionality.

Although by no means a full-fledged Inter-Application Communication (IAC) protocol, HTML combines basic IAC abilities with ease-of-use, network transparency, and a large existing base of media-oriented software components, including web access libraries—features that are desired in today's highly heterogeneous software worlds. For example, *Cecilia*[1] currently offers *Cybil* for providing algorithmic score production, a language similar to, albeit less expressive than Common Music. A production system such as this could benefit greatly from an evaluation mechanism for Common Music services via HTTP as for its *tk*-based graphical editor.

1.6 Teaching Experience

Another goal of the composition server project is to support courses in algorithmic composition. Over the past few years courses at CCRMA and UIUC have resulted in a collection of HTML presentations of algorithmic topics such as randomness, pattern description, chaos, iterative functions and so on. The server

[1] *Cecilia* is a music/sound production system that uses MIT's *Csound* as its sound-processing language (see `http://www.musique.umontreal.ca/Org/CompoElectro/CEC/`).

model allows the functionality of these documents to be extended into interactive, structured sessions with the server, without the necessity for a student to first master the implementation issues of the underlying system. For example, the HTML pages explaining weighted random selection allow the student to experiment with the effects of changing weights in compositional material using a simple table display. When the student selects *GO*, the contents of the table is sent to a corresponding algorithm object in the kernel which renders the choices and returns the results back to the student in the form of a MIDI stream.

2 Implementation

A Common Music system configured to act as a composition server differs from a stand-alone image in several respects. First, it relies on the services of *CL-HTTP*[2], a LISP-based HTTP server, developed and maintained at MIT. Secondly, it contains a new HTTP subsystem that controls the HTTP server and manages distinct user sessions, authenticating HTTP requests and restricting access to features of the Common Music server image as desired. For security reasons, the server image should reside in a private, secured area of the server's filesystem (cf. Figure 1).

2.1 The LISP HTTP Server

Using an internal, LISP-based HTTP server as opposed to a stand-alone, external server has proved to be advantageous for a variety of reasons. Foremost, data that is to be passed back and forth between the server and Common Music need not to be copied, since both share the same application context; also, no data conversion has to take place, since both are LISP software modules. As a result, data exchange between a native, resident server and Common Music is highly efficient, both in terms of speed and memory usage. As a LISP program, the server may also be easier configured and tuned in run-time and under LISP program control by altering its private data structures. This, for instance, makes it easy to add temporary custom MIME type translations or regulate user access to the server. From an administrative point of view, dedicating a separate HTTP server to Common Music permits only these requests to be routed via a different network port, thus contributing significantly to the overall security of the server.

[2] *CL-HTTP* is a full-featured server for the Internet Hypertext Transfer Protocol, written in Common LISP (see `http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html`).

2.2 Session Management

The most difficult problem in making Common Music's services available to a number of clients is the single-user design inherent in the LISP interpreter. However, although concurrent execution is still not part of the upcoming ANSI Common LISP standard, most implementations today support some notion of it and do provide basic facilities to construct a multi-user environment, such as multiple processes and system globals for user home directories. Common Music's HTTP subsystem uses these facilities, when available, to simulate a 'cooperative' multi-user environment that shields user contexts, called *sessions* as much as possible from each other. In particular, by using standard LISP packages and multiple processes, it maintains a different name space, file system area, and process group for each active session.

A session is established, when a user logs in and automatically times out after it has been idle for a configurable time. Once the user has gained authentication, subsequent HTTP requests are automatically assigned to its session, subject to site-specific restrictions. (A Common Music server image running on a remote server, for example, may restrict the use of its serial ports for MIDI to local users only.)

However, evaluation requests submitted via HTTP may contain programming errors or other constructs that bring the LISP system to a halt. As a result, the HTTP subsystem monitors sessions continuously in order to detect, and eventually abort, runaway processes or infinite loops.

Finally, the multi-user environment is also cooperative in the sense that system resources such as memory or disk space are shared among all user sessions, with no concept of quota or priority imposed.

2.3 Security

Lisp is essentially an open-ended development environment. As such, it provides unauthenticated access to the host machine limited only by whatever restrictions the operating system imposes on the user. In fact, part of the power of LISP as a development environment stems from the generality with which it interfaces to traditional operating system services—a generality and ease-of-access that may seriously compromise the host machine's security. On machines where security is of concern, the Common Music Server should reside in a private, separate area of the file system, similar to an anonymous Internet `ftp` server.

When turned into a multi-user environment, LISP is also insecure internally: no read- or write-permissions protect the system from user processes

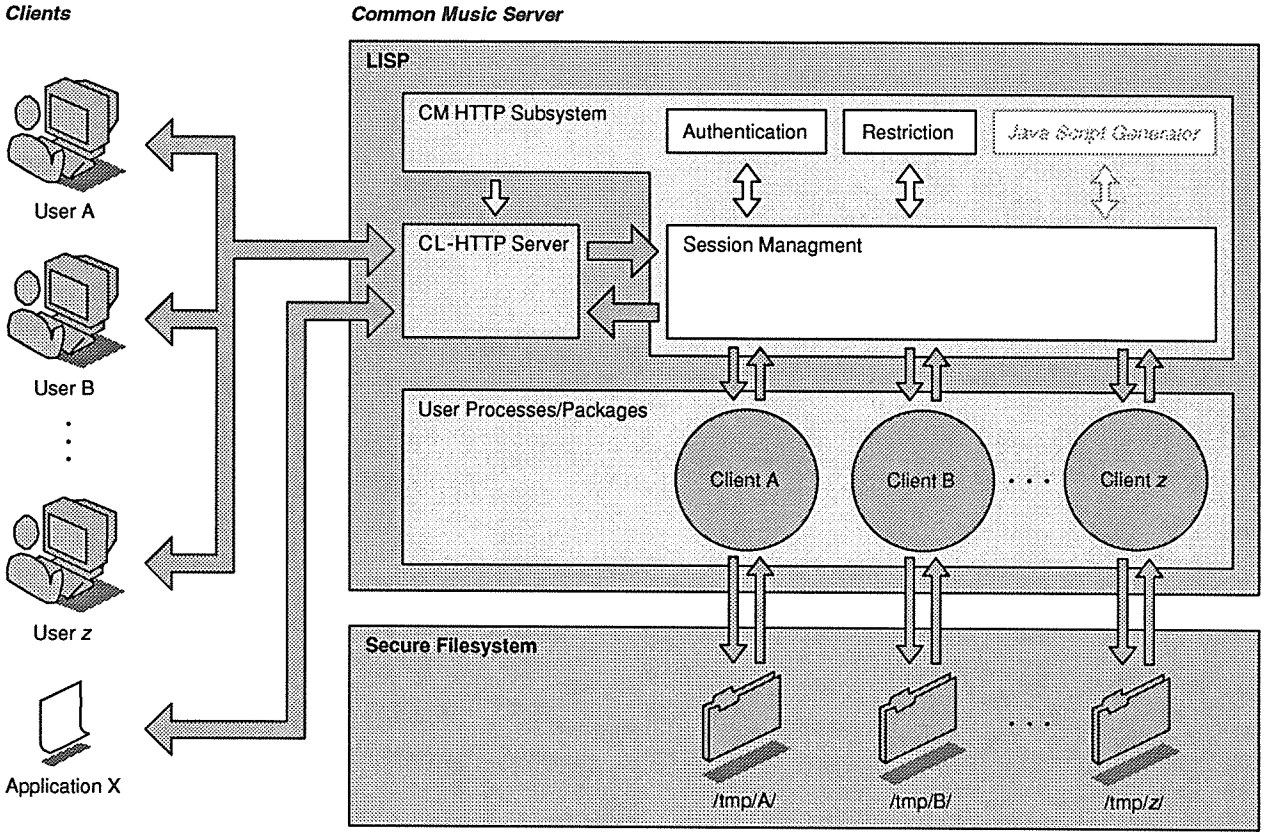

Figure 1: Architecture of Common Music's HTTP Subsystem. A number of users and/or software clients are served concurrently by MIT's LISP-based *CL-HTTP* server software, which is compiled into the Common Music server image. A special HTTP subsystem within Common Music then authenticates requests and manages the various HTTP streams as distinct sessions, allocating a different process group and package name space for each client.

or user processes from affecting each other. Clearly, providing separate name spaces and process groups does not fully protect a session from being affected by others. Specifically, system globals, system packages, or Common Music's toolbox, but also CLOS definitions can not be made read-only without crippling the system and thus seriously jeopardizing its overall usefulness. However, since full session protection within a single LISP system may not be achieved without giving each session a copy of the complete LISP environment—thus essentially reduplicating the LISP image for each user—, they have been left unprotected in the current design. As a result, user processes may affect each other not only actively by invading another user's package, but also inadvertently by side-effecting data in the system area.

2.4 Future Developments

Although the server's internal vulnerability does not appear to have much impact on it's useability in anticipated typical (classroom) situations, it is recognized as a serious design flaw and as a problem affecting the maintenance of the server. A short-term goal is thus to devise a mechanism that adds full write-protection to all system and foreign user packages.

Long-term goals include the addition of a LISP-to-JavaScript translator to the HTTP subsystem to enable remote execution and synchronization of Common Music output such as MIDI and graphics display.

Coupled Mode Synthesis

Scott A. Van Duyne

Center for Computer Research in Music and Acoustics, Stanford University
savd@ccrma.stanford.edu

Abstract

A new way of doing modal-style synthesis has been developed which allows easy control over decay rate profile, natural coupling effects like two stage decay, and much complex timbres for very little additional compute cost, resulting in high quality, *yet* practical, physical modeling of percussion sounds.

1 Motivation

Musical tones, such as plucked or struck strings, bells, plates, drums, wood blocks, etc., may be synthesized in several ways. Additive synthesis uses a bank of oscillators controlled with some set of decaying amplitude envelopes. One difficulty with this method, however, is that the amplitude envelopes may be quite diverse depending on the various mallets or plucking styles to be simulated. Promising work is being done by [2] using neural networks to control the imense amount of control data. On the other hand, interesting physics-based modeling algorithms have been proposed using the digital waveguide mesh [4], which may produce high quality sounds, but are somewhat compute intensive, and will remain so until specialized hardware is developed. The most reasonable and convincing solution till now has been the use of a bank of second-order modal filters. It is relatively low cost; various excitation signals may be employed; and it may be re-struck multiple times producing a natural build up of sound energy. Exciting progress has been made by combining stochastic event modeling with modal synthesis [1].

We propose a reformulation of the modal synthesis method which we call *coupled mode synthesis* (CMS). This approach is based on a traveling wave decomposition of a simple physical model: a group of variously tuned lossless mass and spring oscillators coupled together at some simple, not quite rigid, bridge. In this model, the frequency dependent loss profile of all the modes is efficiently shared within one filter whose parameters are intuitively tweakable. The resonator loops are computed, in some implementations, with one multiply per mode. Since there is only one filter coefficient controlling each modal frequency it is easy to produce independent clickless pitch bending of the modes. Further, real physical mode coupling effects, such as two stage decay and beating, occur naturally with the choice modal tunings and loss filter parameters.

An excitation method is currently being used which is based on that proposed for the commuted piano synthesis algorithm. It is based on a time varying filtering of exponentially decaying noise passed through an effort dependent mallet filter [3]. Realistic percussion sounds have been demonstrated ranging from triangle to orchestral bass drum.

2 Building Up the New Modal Model

2.1 Traditional Modal Synthesis

The sound of a struck or plucked string, plate, bell, block, or other percussive instrument is generally made up of a combination of exponentially decaying sinusoids. Modal synthesis takes note of the fact that these exponentially decaying sinusoids originate from structural vibrating modes in the musical instrument in question; and that these modes can be modeled as a combination of simple second order digital filters of the form,

$$\frac{1}{1 - 2r_k \cos\left(2\pi f_k T\right) z^{-1} + r_k^{\,2} z^{-2}},$$

where the f_k are the modal frequencies (in Hz), the r_k are the attenuations per sample (pole radii) of each mode, and T is the sampling interval ($1/srate$).

These second order resonant filters may be arranged into a bank, tuned and calibrated according to the frequencies and decay rates measured from real sounds and then excited with some suitable excitation signal representing the mallet strike force pulse.

2.2 The Littlest Loop

If we take r_k = 1, that is, the exponentially decaying sinusoid maintains constant amplitude and does *not* delay, then the second order modal filter shown above can be reformulated (plus or minus the addition of a harmless zero in the numerator...) as

$$\frac{1}{1+z^{-1}H_k(z)},$$

where H is a first order allpass filter of the form,

$$H_k(z) = \frac{a_k + z^{-1}}{1 + a_k z^{-1}}.$$

This filter structure corresponds to a tight feedback loop containing three elements: a first order allpass filter, H_k; a single sample delay, z^{-1}; and an inversion, that is, a minus sign at the adder. This structure is shown in Figure 1.

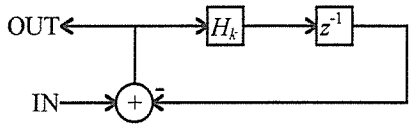

Figure 1: The Littlest Loop

Tuning the Loop The minus sign at the adder inverts the loop signal, which, in the case of a sinusoid, is equivalent to delaying the phase of the sinusoid by half a period, or π radians. The unit delay, z^{-1}, delays the signal further by some small fraction of a radian, say ε. The allpass filter H_k has the capacity to delay the phase of any given frequency, f_k, by from 0 to π radians. By careful choice of allpass coefficient, a_k, we can make it delay the phase of frequency f_k by exactly $\pi-\varepsilon$ radians. This means the total loop round trip phase delay is 2π radians at frequency f_k! The result is that the loop will resonate at that frequency. It is not too tricky to show that the correct coefficient choice is,

$$a_k = -\cos\left(2\pi f_k T\right).$$

2.3 Coupling Little Loops Together

In [3], a method of modeling coupled piano strings is developed. The basic principle is to form single string Karplus-Strong-style loop models containing no internal loss, i.e., they would ring forever all by themselves. Then, two or three of these loops, representing the slightly detuned piano strings in a unison tri-chord group, are coupled together at a nearly rigid, but slightly lossy, bridge impedance. In this way, only one loss filter is needed, and characteristic physical effects such as two stage decay rates and swelling occur naturally with in the coupled filter structure.

In the present case, we have little loops which could be viewed as the smallest possible single mode loops. There is one allpass filter for tuning, and a single sample long "delay line", as shown in Figure 1. We can couple these little tuned loops together at a lumped bridge impedance in the same way as described in [3] for piano string loops, sharing their loss all at one point. Figure 2 shows such a filter structure.

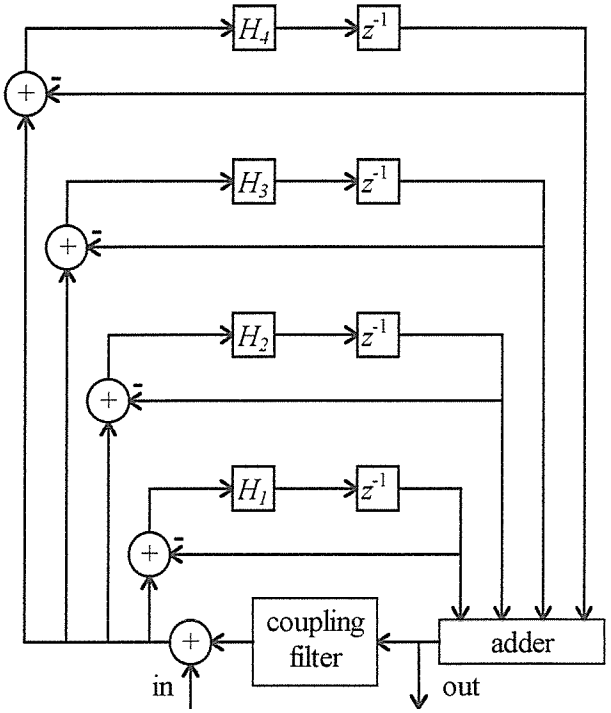

Figure 2: The Coupled Mode Filter Structure

Intuition The way it works can be understood easily on an intuitive level. Referring to Figure 2: Each loop independently oscillates at its specifically tuned frequency; outputs from each loop in the structure are summed together in the "adder"; the adder output signal is then passed through the "coupling filter", which has a very small magnitude response, that is, it attenuates the combined adder output signal greatly. (If the bridge impedance were intended to be perfectly rigid, then the coupling filter would be exactly 0, and no energy would be exchanged among the loops, and each loop would resonate losslessly forever.) In practice, the coupling filter has a magnitude response of something on the or-

der of 0.001 or less. The output of the coupling filter is then added back into each of the oscillating loops; but note that as it is added, it enters at the opposite phase as the loop signal. (That is, at each of the four oscillating loop adders, ⊕, the loop signal it *subtracted*, while the coupling filter output is *added positively*). Hence, the effect of the coupling filter signal being summed back into the oscillating loops at opposite phase is to attenuate each of the loops slightly by a phase cancellation, as well as to slosh the energy around a bit between the loops, leading to the coupling effects.

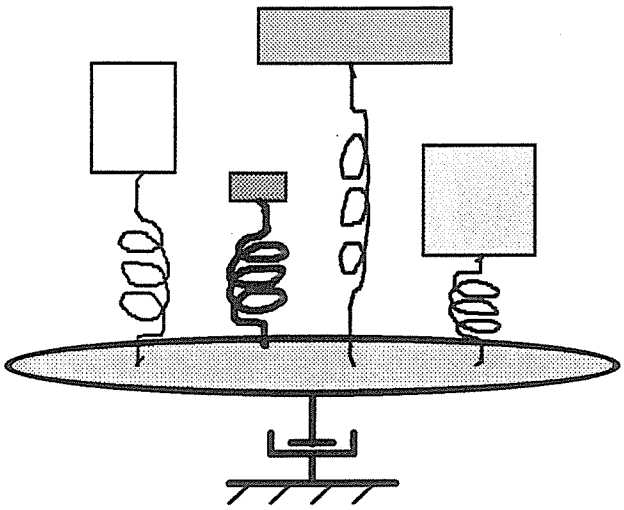

Figure 3: Physical Representation of CMS Algorithm

Figure 3 above, shows a *physical* representation of the coupled mode filter structure in Figure 2. The mass and spring combinations represent single oscillating tuned modes. They are all fixed to a rigid base. Then this base is stuck to the floor via a nearly rigid bridge impedance. With some difficulty, one can show that the filter structure in Figure 2 computes a correct band-limited solution to the system of differential equations implied by the physical system shown in Figure 3.

Calibrating the Coupling Filter While it is easy to tune the oscillating loops, it is a little tricky to see what to do with the coupling filter. However, it seems that an approach like the one described in [3] to calibrate the coupling filter for the coupled piano string model works in this case as well: A theoretical "single mode" loop filter, $L(z)$, may be found whose magnitude response at the various modal frequencies is equal to the per sample attenuation rate of the respective modes; then the coupling filter may be computed from that as,

$$\frac{2(1-L(z))}{1+N+(1-N)L(z)}$$

where N is the number of modes in the coupled mode filter.

Metal *vs.* Wood Since the decay frequency dependent profile of the coupled mode filter is localized in one coupling filter, it is easy to make qualitative modifications to the sound color. It is usual in most materials, that high frequency modes tend to die out more quickly than lower frequency modes. Therefore the L(z) filter described in the preceding paragraph, which represents the frequency dependent decay rate profile, is lowpass in nature, i.e., high frequencies are attenuated faster than low frequencies. Since high frequencies tend to hang around just a little longer in *metal* objects than they do in *wood* objects, it is easy to control the metalic vs. woodlike quality of the sound simply with a single slider which adjusts the amount of lowpass characteristic in the L(z) filter.

2.4 Statistical Modeling of the Modes

In certain kinds of percussive sounds there are many, many densely packed modal frequencies. Having to compute hundreds of modes could make computational requirements of the modal synthesis algorithm impractical. We have found a practical solution to this problem by separating the modes into (1) a small set of psychoacoustically important modes, which may be computed as usual, and (2) a densely packed set of modes, whose exact frequencies are not important, but whose spectral color and overall decay rates are important.

This second set of modes we model statistically, borrowing a concept from the commuted piano soundboard model [3]. The left side of Figure 4 illustrates an enveloped noise model of a general percussive sound with many densely packed modes, and whose higher frequency modes decay somewhat more quickly than its lower frequencies. This is accomplished through the combination of (1) an exponentially enveloped white noise source (which, alone, would represent a densely packed set of modes decaying at exactly the same rate) being feed through a lowpass filter whose bandwidth is being decreased over time, that is, it is becoming more and more lowpassed. The addition of this time varying lowpass filter is to force the higher frequencies in the enveloped noise to decay at a faster rate than the lower frequencies, as is observed in real percussive objects when they are struck.

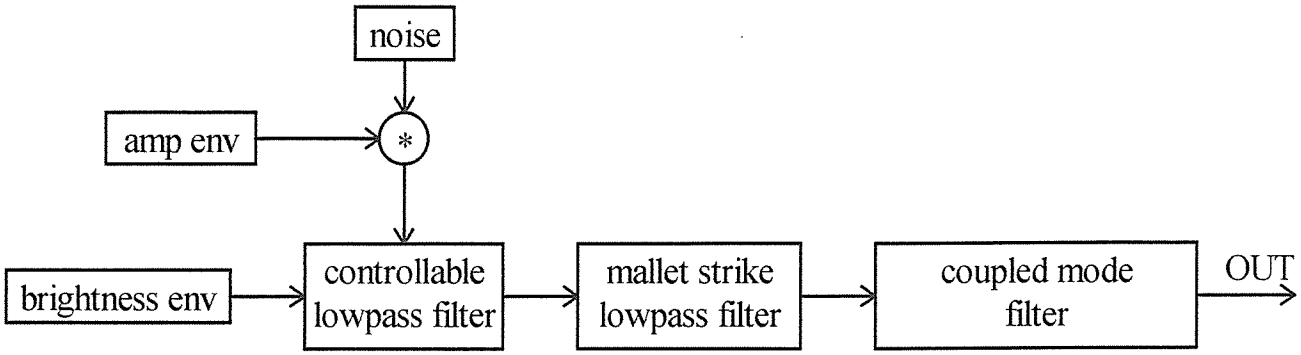

Figure 4: The Full Coupled Mode Synthesis Algorithm

This enveloped noise then represents a statistical model of the densely packed modes. Figure 4 further shows how we pass it through a strike dependent lowpass filter modeling the mallet brightness [3] and onward through the coupled mode filter to add in the more sparse, psychoacoustically important modes.

3 What's so Cool about CMS?

The statistical excitation scheme is cool because it adds a lot of *complexity* to the sound for very little computational cost. Localizing the decay rate profile in one simple coupling filter is cool because it allows simple one-slider controls on overall sound quality (wood vs. metal, for example); therefore easy *morphing* strategies are possible. Coupling the modes together at a single lumped impedance is cool because *two stage decay* phenomena and beating decay envelopes come for free in a natural way. Sharing the loss at one point is cool because, in some implementations, it *can save overall compute cycles*. Formulating the oscillating loops with first order allpass filters is cool because it reduces the pitch control to one filter coefficient *making pitchbend effects easily controllable*. CMS is also cool because it is relatively *straightforward to calibrate* to existing sounds, and at the same time, relatively *easy to experiment* with, just moving the sliders around.

4 Acknowledgments

Funding for the development of the CMS algorithm and the calibration of a set of useful sounds was provided by Stanford University through the Office of Technology Licensing as part of the Sondius® trademark development program, in collaboration with the Center for Computer Research in Music and Acoustics. Thanks to Tim Stilson, who efficiently implemented the CMS algorithm on the Motorola 56000 DSP and who assisted in the development of calibration "trending" software. Thanks to Nick Porcaro and Pat Scandalis, who integrated Tim's DSP code into the SynthBuilder™ development environment., added useful high level features, and assisted in the calibration of many sounds. And a special thanks to Dr. John R. Pierce, whose collaboration with me on the Passive Nonlinear Filter project in 1993 inspired me to respect the power of the first order allpass filter and to search for other of its non-obvious applications.

References

[1] Cook, P. 1996. "Pysically Informed Sonic Modeling (PhISM): Percussive Synthesis," *Proc. ICMC*, Hong Kong.

[2] Freed, A.; M. Goldstein; M. Goodwin; M. Lee; K. McMilleml; X. Rodet; D. Wessel; and M. Wright. 1994. "Real-Time Additive Synthesis Controlled by a Mixture of Neural-Networks and Direct Manipulation of Physical and Perceptual Attributes," *Proc. ICMC*, Aarhus.

[3] Van Duyne, S.; and J. Smith. 1995. "Developments for Commuted Piano," *Proc. ICMC*, Banff.

[4] Van Duyne, S.; and J. Smith. 1993. "Physical Modeling with the 2-D Digital Waveguide Mesh," *Proc. ICMC*, Tokyo.

A Lossless, Click-free, Pitchbend-able Delay Line Loop Interpolation Scheme

Scott A. Van Duyne
David A. Jaffe
Gregory Pat Scandalis
Timothy S. Stilson

Center for Computer Research in Music and Acoustics, Stanford University
savd@ccrma.stanford.edu, daj@ccrma.stanford.edu,
gps@stanford.edu, stilti@stanford.edu

Abstract

An efficient method for signal controllable fractional delay implementation has been found. It combines the flexible control of linear interpolation with the frequency independent losslessness of allpass interpolation, avoiding the undesirable effects of each method.

1 Problem Statement

The development of high quality, well calibrated, physical modeling synthesis algorithms based on Karplus-Strong-style feedback loops [3][2] requires the use of delay lines with non-integer lengths. Ideally, these delay lines should have two important features:

(1) Their lengths must be smoothly and fractionally variable by some control signal in order to implement pitch bend, glissando, and vibrato effects.

(2) They must be lossless at all frequencies to minimize unwanted decay in physical modeling feedback loop structures.

Unfortunately, no standard methods for interpolation of non-integer length delay lines are in general use which have *both* of these required features. On the one hand, non-integer length delay lines that use *linear interpolation*, or other FIR interpolation methods, can be varied smoothly in length by a control signal, but they have unsatisfactory energy losses caused by the FIR interpolation filter itself in the high frequency region and, in particular, in high pitched loops. This causes high pitched musical notes to decay away too quickly.

On the other hand, standard *allpass interpolation*, known since the beginnings of Karplus-Strong-style string modeling [2], solves the energy loss problem for the fixed pitch case. However, when implementing pitch bend, glissando, or vibrato effects, allpass filters introduce undesirable artifacts, such as audible clicks. This is primarily due to the internal state in the recursive allpass filter, which must be handled carefully when changing the filter coefficient. Until now, the practical choice has been between allpass interpolation for high frequency sustainability on the one hand, and linear interpolation for flexibility of pitch bend control on the other.

We have formulated a new delay line interpolation structure which has the time-varying delay length flexibility of simple linear interpolation, while retaining the energy conserving effects of fixed allpass interpolation. This was achieved by combining previous results from three different quarters: a smooth waveguide legato implementation crossfading trick [1], some initial results in click reduction in time varying fractional allpass interpolation [6], and a few psychoacoustical observations about just noticeable differences in pitch [5].

2 Puzzle Pieces

2.1 Linear Interpolation

For good tuning of a Karplus-Strong loop, the delay line length must be equal to:

$$DelayLength = SamplingRate\ Frequency$$

This generally comes out to a non-integer number of samples, and rounding to the nearest integer is just not good enough at current sampling rates, e.g., 44.1 kHz. It is easy to delay a signal by 25 samples, but delaying a signal by 25.3 samples is problematic in a sampled system since 0.3 samples is undefined!

Linear interpolation solves the problem by taking a weighted average of the two closest delay lengths. The

following linear interpolation, illustrated in Figure 1, implements a delay of 25.3 samples:

$$OUT(n) = 0.7 \times IN(n-25) + 0.3 \times IN(n-26)$$

The problem with linear interpolation is that high frequencies tend to get wiped out quickly with repeated averaging. Remember that the original Karplus-Strong plucked string algorithm [3] called for a two point average filter in a delay line feedback loop, similar to the equation above. The purpose of the averaging filter was to smooth the waveform a little bit each period, thereby forcing the high frequencies to die away faster than the low frequencies, and to create the qualitative effect of a plucked string decay. This effect is very strong for high pitched loops since the delay line part is short and the loss is greater for high frequencies. This means getting a good long sustain on a Karplus-Strong-style high guitar string sound is impossible (unless you don't care if it is in tune!).

On the other hand, the linear interpolation approach is very flexible. If the you want to change the delay length continuously, as in bending the pitch of a note in the plucked string model, then just slide the linear interpolator along the delay line adjusting the sample weights appropriately.

More extravagant weighted averages, known as FIR interpolation methods, take advantage of more than just two adjacent samples and can improve the frequency response of the interpolation some, but also are increasingly more difficult to compute. Laakso et al. give a comprehensive review of both FIR and allpass interpolation design methods in [4].

2.2 Allpass Interpolation

Allpass interpolation solves the problems of high frequency energy loss in feedback loops which are presented by linear interpolation and FIR schemes. It trades error in the magnitude response, which causes unwanted decay in the high frequencies, for error in the phase response, which only causes incidental detuning of the highest partials. Allpass interpolation in the context of Karplus-Strong models was first proposed by Jaffe and Smith [2] in 1983. They noted that, for a desired fractional delay of d samples, an allpass coefficient of

$$a \approx (1-d)/(1+d)$$

could be chosen as a reasonable approximation.

The following allpass interpolation scheme, illustrated in Figure 2, implements a delay of 0.3 samples:

$$OUT(n) = a \times x(n) + x(n-1) - a \times OUT(n-1)$$

where, $a \approx (1-0.3)/(1+0.3) \approx 0.5385$.

Note that allpass interpolation is recursive; that is, the interpolation uses not only a combination of input samples, $x(n)$ and $x(n\text{-}1)$, but also adds in part of its previous output sample, $OUT(n)$. If you continuously change the coefficient, a, to create a pitch bend or glissando effect, then very special attention must be paid to correcting for the recursive effect if you want to avoid clicks and glitches in the sound as you change pitch [6]. Furthermore, the problem of what to do when you change the integer part of the delay line length, as well as the fractional allpass interpolation part, is an other complicated problem.

Minimizing the transient effect The discontinuity resulting from changing the coefficient, a, can be minimized by keeping the coefficient value as close to zero as possible. The transient effect of changing the coefficient rings out at a rate proportional to the series: a, a^2, a^3, ... We note that if the delay, d, is kept within the unit range, 0.618 to 1.618, then the coefficient, a, remains between -0.236 and +0.236. This means that, with d in this range, the transient effect after 5 samples is a maximum of $(0.236)^5$ or about 62 dB down. In effect, the allpass interpolation filter may be held to a 5 sample warm up time. [6] makes a similar observation.

2.3 The Legato Crossfade Trick

Another piece of the puzzle is a solution to the problem of producing legato transitions between tones of different pitch using the a single feedback loop. If you change the delay length suddenly there is a click in the sound. If you gradually glide the delay length from the first value to the second, using, for example the sliding weighted average linear interpolation method described above, you will hear and unwanted glissando effect, rather than a legato effect.

A legato crossfade method is described briefly in [1] in the context of legato commuted synthesis violin bowing. The first part of the legato trick was the use of a *circular buffer* delay line implementation. Circular buffer simply means that the delay line is implemented in a large fixed length piece of memory with a read pointer chasing a write pointer around, always the appropriate number of samples behind it. Although the actual delay length is shorter than the full length of the memory being used, nevertheless, the full memory is filled with similar looking waveform. That is, as the write pointer progresses through the cicular buffer memory, it lays

down perfectly good waveform at the currently specified pitch throughout the full memory buffer.

Jaffe noted that if you simply introduce *two* read pointers, one set at the delay of the *first* note, and the second set for the delay of the *second* note, then you can crossfade between the two read pointers, over the course of about 15-30ms, to produce a very realistic legato, *not glissando*, effect. This structure is illustrated in Figure 3. There is no glitch in the tone since the full memory buffer is filled with reasonable looking waveform at the current pitch. Therefore, the *second* read pointer is looking are perfectly good data initially and the crossfade is gradual. Stilson developed a similar trick, independently, for use in a pitch shifting algorithm.

3 Glissable Allpass Interpolation

By combining elements of the interpolation and legato methods described above, we can find a practical structure for a flexible lossless fractional delay line. The basic inspiration is this: Let's view glissando as a lot of very fast, tiny, legato transitions. Start with a circular buffer delay line with two *allpass interpolated* readers. Then send new fractional delay length values to the alternating allpass interpolated readers every 16 samples, for example.

What is the problem with this? The allpass interpolation filters will be producing clicks every time a reader is set to a new position! *But* the transient effect lasts only 5 samples if the fractional delay range is maintained between 0.618 and 1.618! The 5 warm-up can be ignored by using a special crossfading function which waits 5 samples before crossfading over to the newly set allpass interpolated reader. When using a 16 sample alternation rate, this leaves 11 samples to do the actual crossfade, which, in practice, is enough. See Figure 4.

Psychoacoustical Detail The human hearing system is only able to detect a finite number of different pitches. Two tones which are sufficiently close together in pitch become indistinguishable. There is a *just noticeable difference* (JND) threshold for human pitch differentiation. The number of JNDs per octave varies with the register, but a representative worst case for us is that there are about 280 JNDs between 1000 Hz and 2000 Hz (or, approximately, between c6 and c7) [5]. A JND comes out to about 0.1 samples in a Karplus-Strong feedback loop delay length for a 1000 Hz tone being computed at a sampling rate of 44.1 kHz. It is easy to show that running the alternating crossfader at a tick rate of once per 16 samples, and using a maximum glissando rate of one JND per tick, that we can gliss an octave from c6 to c7 in about one tenth of a second $(280 \times 16 / 44100 \approx 0.1)$.

DSP Implementation Trick Many fixed-point DSP chips can perform very fast multiply-add operations but do *not* support a fast divide operation; so, computing the allpass coefficient,

$$a \approx (1-d)/(1+d),$$

every 16 samples is actually rather inconvenient. Fortunately, we may expand the expression in a Taylor Series about the point, $d = 1$, giving,

$$a \approx -\frac{(d-1)}{2} + \frac{(d-1)^2}{4} - \frac{(d-1)^3}{8} \cdots$$

a very efficient computation using only multiplies, adds and, possibly, right shifts. Maximum error in three terms is 0.024 samples. Recall a JND at 1000Hz is about 0.1 samples.

4 Acknowledgments

Funding for this work was provided by the Stanford University through the Office of Technology Licensing as part of the Sondius® trademark development program, in collaboration with the Center for Computer Research in Music and Acoustics.

References

[1] Jaffe, J., and J. Smith. 1995. "Performance Expression in Commuted Waveguide Synthesis of Bowed Strings," *Proc. ICMC*, Banff.

[2] Jaffe D., and J. Smith. 1983. "Extensions of the Karplus-Strong Plucked String Algorithm," *Computer Music Journal*, Volume 9, Number 2.

[3] Karplus, K., and A. Strong. 1983. "Digital Synthesis of Plucked-String and Drum Timbres," *Computer Music Journal*, Volume 7, Number 2.

[4] Laakso, T.; V. Valimaki; M. Karjalainen; and U. Laine. 1996. "Splitting the Unit Delay," *IEEE Signal Processing Magazine*, January.

[5] Olson, H. 1967. *Music, Physics, and Engineering*, Dover Publications, Inc.

[6] Valimaki, V.; T. Laasko; and J, Mackenzie. 1995. "Elimination of Transients in Time-Varying Allpass Fractional Delay Filters with Application to Digital Waveguide Modeling," *Proc. ICMC*, Banff.

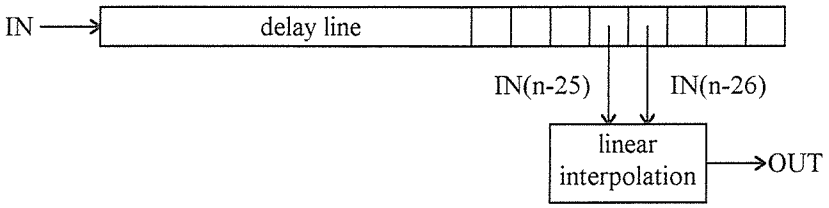

Figure 1: Flexible Linear Interpolation

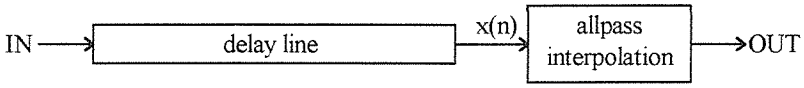

Figure 2: Fixed Allpass Interpolation

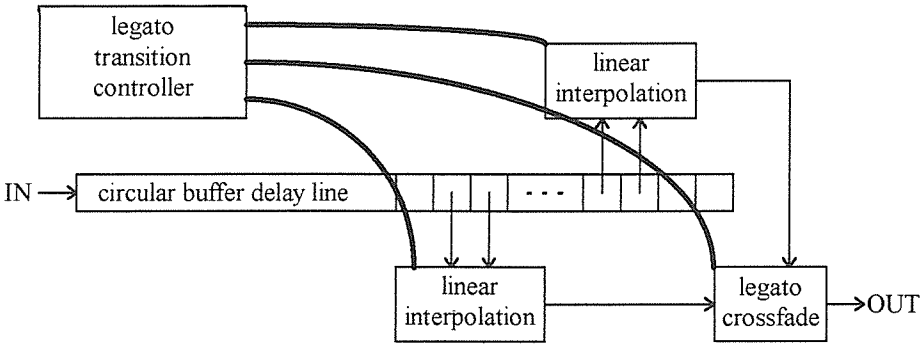

Figure 3: Legato Transition Trick

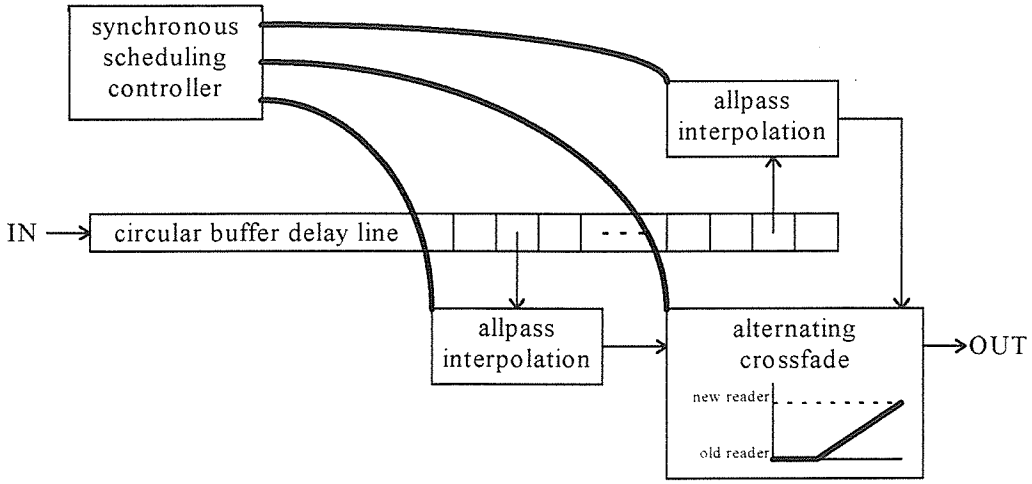

Figure 4: Glissable Allpass Interpolation

Transient Modeling Synthesis: a flexible analysis/synthesis tool for transient signals

Tony S. Verma (1), Scott N. Levine (2), Teresa H.Y. Meng (1)
(1) Department of Electrical Engineering, Computer Systems Laboratory, Stanford University
(2) Center for Computer Research in Music and Acoustics (CCRMA), Stanford University

Abstract

We propose a flexible analysis/synthesis model for transient signals that effectively extends the Spectral Modeling Synthesis (SMS) parameterization of signals from sinusoids+noise to sinusoids+transients+noise. The explicit handling of transients provides a more realistic and robust signal model. The model presented is a parametric model for transients that allows for a wide range of signal transformations. In addition to modeling, a transient detection scheme is also presented.

1 Introduction

Transient Modeling Synthesis (TMS) is a flexible analysis/synthesis tool for transient signals. TMS is the frequency domain dual to sinusoidal modeling. Sinusoidal modeling and Spectral Modeling Synthesis (SMS) [1, 2] have enjoyed a rich history in both speech and audio. SMS is a flexible signal model that consists of sines and noise. While SMS provides a representation for sinusoidal signals that allows a wide range of transformations, TMS provides a representation for transient signals that allows a wide range of transformations. TMS combined with SMS effectively extends the sines+noise model to sines+transients+noise. The explicit handling of transients provides a more robust signal model and is essential for synthesizing realistic attacks of many instruments. Although there has been work on explicit handling of transients within the SMS framework [3, 4], these methods are not flexible in their representation of transients. TMS not only allows explicit handling of transients, but allows manipulation of the model parameters and thus maintains the spirit of SMS as a flexible signal representation. The first section of the paper describes the framework of TMS. The second describes a transient detection scheme that allows TMS to work more effectively. The final section gives an analysis/synthesis example.

2 The TMS Framework

An explicit transient model is motivated because transients do not fit well into the SMS framework. SMS is a parametric modeling tool that consists of two parts: sinusoidal modeling and noise modeling. The analysis portion of sinusoidal modeling finds well developed sinusoids by tracking spectral peaks over time. It finds the sinusoidal components in a signal by using short-time Fourier analysis and tracking meaningful peaks from frame to frame. During synthesis, these meaningful peaks, which consist of the parameter triplet $\{magnitude, frequency, phase\}$, control a bank of oscillators (additive synthesis) or can be used in an inverse Fourier Transform/overlap add scheme for signal reconstruction. SMS furthers its decomposition by considering a residual signal. This residual signal is the difference between the original signal and the synthesized sinusoidal signal. The residual consists of components that are not well modeled by sinusoids. These components are transients and noise [3, 5]. In the SMS framework, the transient+noise residual is modeled as slowly varying filtered white noise. Transients in the residual do not fit within this model. This is a serious drawback when considering instruments with sharp attacks because transients modeled as noise become smeared in time and the attack is lost.

As suggested by others [3, 6], transients need to be considered separately from noise. Others have done this by removing transient areas from the residual, performing noise analysis, then adding the transients back into the signal. Although this method works, it has a few drawbacks. First, it lacks flexibility in representing transients. Representing transients as PCM samples is far from the flexible representation goal of SMS. Secondly, many instruments have an underlying noise, the breathiness of a flute

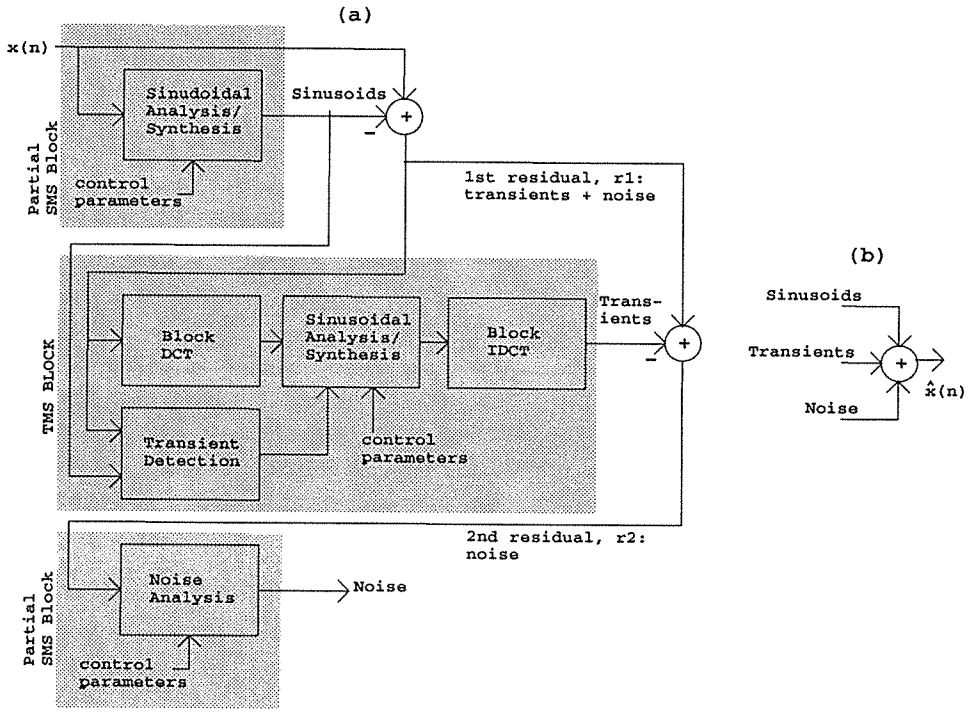

Figure 1: (a) Analysis block diagram. (b) Synthesis block diagram

for example, that is neither sinusoidal or transient. When removing transients in the fashion stated, both transients and noise are removed. It is desirable to model transients separately but leave noise to the noise model. These needs motivated TMS.

The system block diagram for the combination of TMS and SMS is given in figure 1. We use TMS on the first residual, $r1$, which consists of noise and transients. TMS optionally first detects where transients occur. It then fits a parametric model to the transients. The transients are synthesized and subtracted from the first residual, $r1$, to create a second residual, $r2$. This second residual consists of primarily slowly varying white noise. Thus attacks of instruments are well preserved and the underlying noise of an instrument is left for the noise model.

Since sinusoidal modeling tracks well developed sines, it cannot track transients which are time-limited, pulse-like signals. However, pulse-like signals in one domain can be periodic in another domain. The basic idea underlying TMS is the duality between time and frequency. TMS is the frequency domain dual to sinusoidal modeling. While the analysis portion of sinusoidal modeling finds sinusoids by tracking the well developed spectral peaks of a time domain signal, TMS finds transients by tracking the well developed spectral peaks of a frequency domain signal. That is, we first map segments of the time domain signal into the frequency domain. This causes transients in the time domain to become periodic in the frequency domain. We then perform sinusoidal modeling on this frequency domain signal. The well developed spectral peaks found from analyzing the frequency domain signal represent well developed transients in the original time domain signal. The block length of the time to frequency domain mapping must be sufficient to make transients compact entities within the block. A block size of about one second is sufficient.

The mapping from the time domain to frequency domain is chosen so that transients in the time domain become sinusoidal in the frequency domain. The Discrete Cosine Transform (DCT) provides such a mapping. It is defined as:

$$C(k) = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)k\pi}{2N}\right]$$
$$\text{for } n, k \in 0, 1, \ldots, N$$

$$\alpha = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } k = 1, 2, \ldots, N \end{cases}$$

Therefore if $x(n)$ is a Kronecker delta, then $C(k)$ is a cosine whose frequency depends on the time location of the impulse. Roughly speaking, an impulse that occurs toward the beginning of the frame results in a DCT domain signal that is a relatively low frequency cosine. If the impulse occurs toward the end of the frame, then the DCT of the signal is a relatively high frequency cosine.

For signals commonly encountered, attacks are rarely a simple Kronecker delta, but are usually groups of closely spaced samples. This results in a DCT that consists of many closely spaced sinusoids or a sinusoid that varies slowly over the domain of definition of the DCT. This type of periodic signal is exactly what sinusoidal modeling works well on. Thus we obtain a parametric model of transients by performing sinusoidal modeling on the DCT of blocks of the time domain signal.

In order to perform sinusoidal modeling on the DCT of a section of a signal, we must take overlapping discrete Fourier transforms (DFT) on the DCT domain signal. This combination of operations, DCT then DFT, brings the signal back into some type of time-like domain. Although this may seem redundant, these operations rotate (unitary transforms simply rotate vector spaces) the signal in such way to make transients readily apparent.

When performing short-time Fourier analysis in the DCT domain, the window size for the spectral analysis must be shorter than the DCT size, but adequate zero-padding or other frequency interpolation methods, such as parabolic interpolation [3] must be used to avoid quantization of time. Each spectral peak found in the DCT domain is a triplet of information, as in the case of sinusoidal modeling, of the form $\{magnitude, frequency, phase\}$. This *frequency* parameter, however, is actually time domain information. The *frequency* here corresponds to where a transient occurs. This dictates how much frequency resolution (actually time resolution) is required when performing TMS. Since *frequency* corresponds to a time location, we must guarantee that the amount of frequency resolution is greater than the number of time samples used in computing the block DCT in order to avoid quantization of transients.

3 Transient Detection

If we know where possible transients occur, we can restrict TMS to model transients only in those areas. This is done by keeping only those frequencies in the TMS domain that occur in possible transient areas. This allows TMS to run more efficiently because peaks that are clearly not in transient areas are not given to the spectral tracking algorithm. This step is optional because running TMS without restrictions and proper control parameters can be reliably used to find onsets. We describe here, however, a simple transient detection scheme based on energy in the synthesized sinusoids, denoted s, and the first order residual, $r1$.

Qualitatively, possible transients occur when the sinusoidal model breaks down and the energy in the first order residual increases rapidly. To quantitatively measure this, we look at energy in s and $r1$ over the entire DCT block and over a smaller short-term sliding window. Let

$$E_x = \sum_{n=0}^{N-1} |x(n)|^2$$

denote the energy of a signal over the DCT block where N is the length of the DCT. Then E_s and E_{r1} denote the energy in s and $r1$, respectively, over the DCT block length. Now define the energy in the sliding window as:

$$e_x(k) = \sum_{n=k-\frac{L}{2}}^{k+\frac{L}{2}} |x(n)|^2 \quad \text{for } k = \alpha 1, \alpha 2, \ldots \qquad (1)$$

Where L is the length of the sliding window, α is the hop size and x is properly defined, e.g. zero padded, outside of the region $n = 0, 1, \ldots, N$ so equation 1 makes sense at the edges of the DCT block. A possible transient occurs when the ratio of normalized short-term energy of $r1$ and s is larger than some threshold. Specifically, when

$$\frac{e_{r1}(k)/E_{r1}}{e_s(k)/E_s} > THRESHOLD \qquad (2)$$

possible transient areas are noted and frequencies in the TMS domain outside of these areas can be discarded.

4 Examples

As an example, we show the sines+transients+noise analysis on a xylophone hit, the results of which are shown in figure 2. The xylophone, although inharmonic, has a perceived pitch which is modeled well by the sine portion of the representation. Figure 2(a) is a plot of the original signal sampled at $44.1KHz$, while figure 2(b) shows the synthesized sinusoids. Figure 2(c) is the first residual, $r1$, which shows the sharp attack of the sound as well as some underlying noise. The attack, as modeled by TMS, is shown

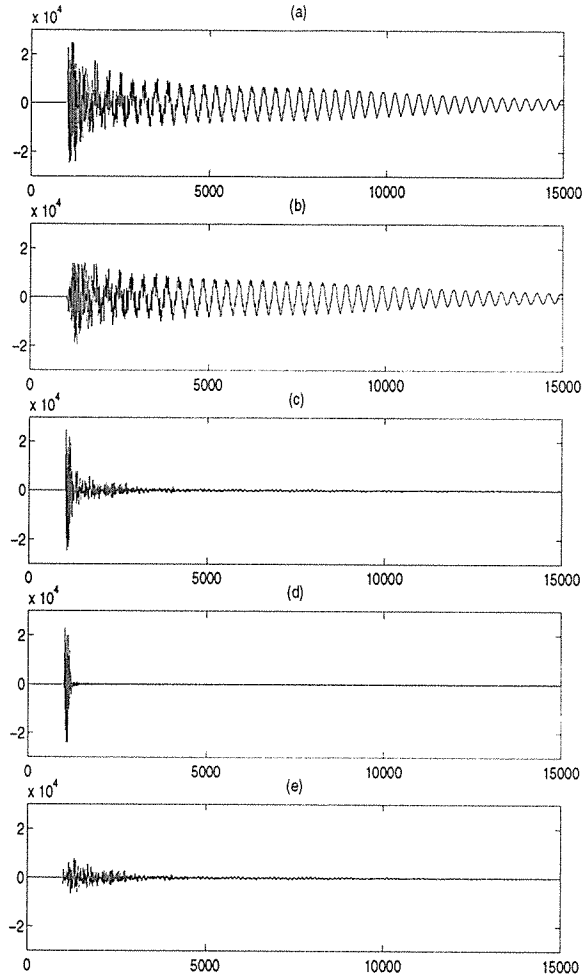

Figure 2: (a) Original xylophone. (b) Synthesized sinusoids. (c) First residual containing transients+noise. (d) Synthesized transients. (e) Second residual containing noise

in figure 2(d). Figure 2(e) shows the second residual, $r2$, which is the part of the original signal that is not well modeled by sines or transients. This is slowly varying noise. If the first residual signal were passed to the noise model without TMS, the attack would be smeared and the characteristic 'knock' of the xylophone would be lost. The summation of the sines+transient+noise portions yield a signal that is perceptually indistinguishable from the original.

5 Conclusions

There are many benefits to using TMS. First, synthesized attacks of instruments are well preserved while maintaining a flexible model of these attacks. Combining TMS with SMS allows modeling of a wide range of sounds while allowing the synthesized versions to be perceptually identical to the original. In addition, because noise models assume residual signals which consist of slowly varying noise, using TMS to remove transients allows the models to work more effectively. Finally, because TMS has the same flexibility as SMS, a large number of transformations are possible on the analyzed signal. In addition, these transformations will be more robust because we have an explicit parameterization for sines, transients and noise. For example, when time stretching a signal, it is desirable for transients to move to their proper onset locations but remain localized, while the harmonics and noise stretch. By using TMS combined with SMS, these types of transformations are possible. Many other signal transformations are a subject of current research.

References

[1] Xavier Serra and Julius O. Smith, "Spectral modeling synthesis: a sound analysis/synthesis system based on a deterministic plus stochastic decomposition", *Computer*, vol. 14, no. 4, pp. 14–24, WINTER 1990.

[2] Robert J. McAulay and Thomas F. Quatieri, "Speech analysis/synthesis based on a sinusoidal speech model", *IEEE Transactions on Acoustics Speech and Signal Processing*, pp. 744–754, 1986.

[3] Xavier Serra, *A System For Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition*, PhD thesis, Stanford University, 1989.

[4] K. N. Hamdy, M. Ali, and A. H. Tewfik, "Low bit rate high quality audio coding with combined harmonic and wavelet representations", *Proceedings of ICASSP-96*, vol. 2, pp. 1045–1048, May 1996.

[5] E. Bryan George and Mark J. T. Smith, "Analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones", *J. Audio Engineering Society*, vol. 40, no. 6, pp. 497–515, June 1992.

[6] Michael Goodwin, "Residual modeling in music analysis/synthesis", *Proceedings of ICASSP-96*, vol. 2, pp. 1005–1008, May 1996.