CENTER FOR COMPUTER RESEARCH IN MUSIC AND ACOUSTICS
OCTOBER 1992

Department of Music
Report No. STAN-M-80

PHYSICAL MODELING AND SIGNAL PROCESSING
CCRMA PAPERS PRESENTED AT THE
1992 INTERNATIONAL COMPUTER MUSIC CONFERENCE

Perry R. Cook, Brent Gillespie, Julius O. Smith
Atau Tanaka, Scott Van Duyne

CCRMA
DEPARTMENT OF MUSIC
Stanford University
Stanford, California 94305

CCRMA Papers on Physical Modeling and Signal Processing from the 1992 International Computer Music Conference

Perry Cook: A Meta-Wind-Instrument Physical Model, and a Meta-Controller for Real-Time Performance Control

Brent Gillespie: Dynamic Modeling of the Grand Piano Action

Brent Gillespie: Touch Back Keyboard

Smith and Cook: The Second-Order Digital Waveguide Oscillator

Tanaka: Implementing Quadraphonic Audio on the NeXT: Hardware and Software Issues

Van Duyne and Smith: Implementation of a Variable Pick-Up Point on a Waveguide String Model With FM/AM Applications

Scott A. Van Duyne: Low Piano Tones: Modeling Nearly Harmonic Spectra With Regions of FM

A meta-wind-instrument physical model, and a meta-controller for real time performance control

Perry R. Cook

Stanford CCRMA, Stanford, CA, 94305
PRC@CCRMA.STANFORD.EDU

ABSTRACT

A waveguide instrument has been constructed which simulates the main acoustical mechanisms of the wind instrument family, allowing one instrument to function as flute, recorder, clarinet, saxophone, trumpet, trombone, and other instruments which are hybrids of the various wind instrument sub-families. Multiple voices of the instrument can be run in real-time on a Motorola DSP 56001 signal processing chip, under control from a NeXT scorefile, mouse control with graphical feedback on the NeXT computer screen, or via MIDI. The instrument is controlled by parameters affecting reed stiffness, mass-spring-damper control of the reed/lip oscillator, length of the bore and of secondary delay paths, state of tone/register holes, jet length, and breath pressure. Lower level control allows direct access to control coefficients, allowing hybrid instrument settings to be defined.

A special MIDI controller has been constructed for controlling this instrument as well as more conventional MIDI synthesizer instruments. The controller allows a musician to play in the paradigms of most wind instruments, but provides all of these modes and others simultaneously. Controls include linear slide control like that of the trombone, valve control like that of other brass instruments, fingering control like that of the woodwinds, breath pressure, lip tension, bite pressure, and rotational position (similar to rotating the flute or head to change embouchure). A new rotational control is mounted in the mouthpiece which varies playing position from recorder-like (end blown) to flute-like (edge blown). The controller allows a performer to effectively use much more of the real time control bandwidth available in the mouth and hands.

1. SIMPLE WAVEGUIDE INSTRUMENTS FOR REAL-TIME SYNTHESIS

Synthesis of musical sounds by physical modeling is becoming more feasible due to two components: computing equipment is growing increasingly powerful and inexpensive, and more efficient algorithms for computing the physical solutions are emerging [Smith87]. Once the basic physics of an instrument or instrument family are understood, a computational model can be derived, and that model can be simplified for more efficient computation. Optimization often takes advantage of the idiosyncrasies of specialized DSP hardware. Figures 1-3 show three simplified physical models, based loosely on the physics of brass instruments, jet-reed instruments, and cane reed instruments. Non-linearities are modeled as polynomials for efficient DSP calculation.

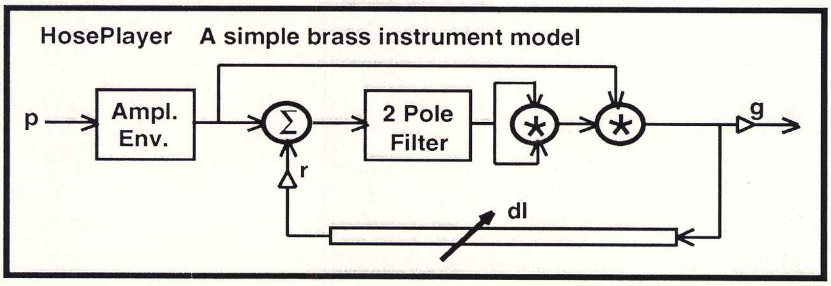


Figure 1. A simple physical brass wind instrument model for DSP implementation.

The brass instrument model shown in Figure 1 uses a second-order digital filter to model the mass-spring-damper oscillator system of the lip [Cook91]. This filter solves for the position of the lip as a function of the differential pressure applied by the mouth and bore. The pressure input to the bore is found by squaring the lip position, and multiplying by the input breath pressure. The bore is modeled by a delay line of length *dl*, and an asymptotic envelope generator provides smoothing of the instantaneous breath input pressure *p*. The variable *g* is an output scaling parameter, and *r* is the net reflection gain. The instrument as shown requires a total of 7 multiplies and 3 adds per sample computation. As with all instrument models in this paper, memory useage is dominated by the delay line length(s).

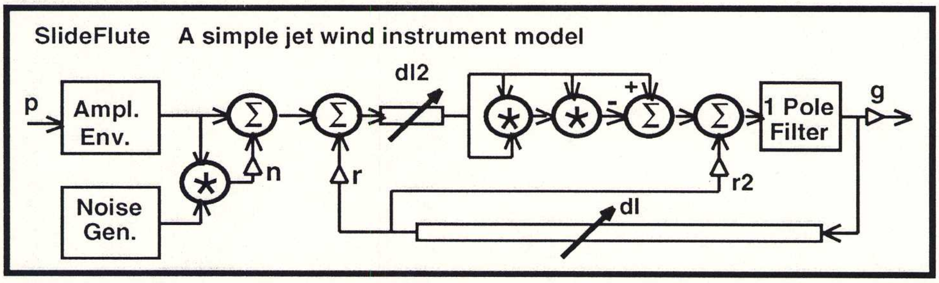


Figure 2. A simple physical jet-reed wind instrument model for DSP implementation.

The jet-reed instrument model shown in Figure 2 approximates the sigmoidal non-linearity of the jet using the polynomial $x - x^3$. The second delay line of length *dl2* models the propagation time of the jet reed, as discussed by Fletcher [Fletcher91] and implemented by Karjalain and others [Karjalain91]. A 1-pole filter models the low-pass reflection function at the end of the instrument. The variable *n* determines the amount of random noise mixed into the breath pressure excitation function. The variables *dl*, *r*, *g*, and *p* perform the same functions as in the brass instrument model. The instrument as shown requires a total of 9 multiplies and 6 adds per sample computation.

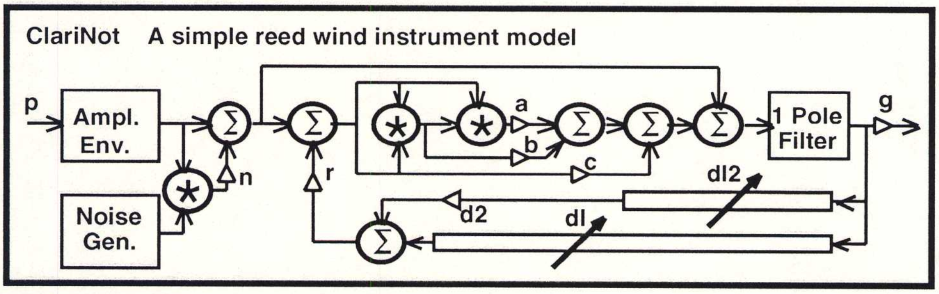


Figure 3. A simple physical reed wind instrument model for DSP implementation.

The reed instrument model shown in Figure 3 allows the use of an arbitrary polynomial of the form: $ax^3 + bx^2 + cx$ for the non-linearity of the reed. The second delay line models the reflection effects of a register hole, suppressing the fundamental and emphasizing a particular harmonic determined by the length variable *dl2*. A 1-pole filter models the low-pass reflection function at the end of the instrument. The variables *dl, r, g, n,* and *p* perform the same functions as in the brass instrument model. The instrument as shown requires a total of 12 multiplies and 8 adds per sample computation.

2. WHIRLWIND: A META-WIND INSTRUMENT PHYSICAL MODEL

Figure 4 shows the result of combining the models of Figures 1-3, yielding a model which can simulate simple reed, jet, and lip excitation models. Pitch control paradigms of tube length, register hole, jet length, and embouchure are available. The instrument requires a total of 20 multiplies and 17 adds per sample computation.

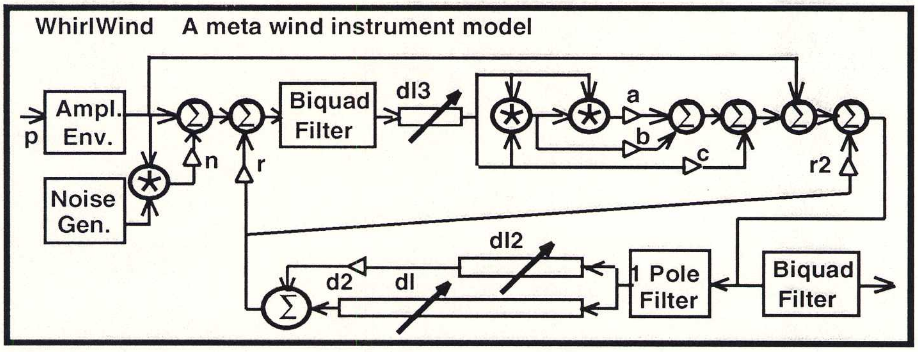


Figure 4. WhirlWind: the meta-wind instrument physical model.

3. HIRN: A NEW REAL-TIME SYNTHESIZER CONTROLLER

Based on the premise that wind players have much unused control bandwidth available while playing their instruments, a Meta-Wind Instrument Controller was designed and constructed. Called HIRN, this controller exploits the common control modes found in the wind instrument family, and adds many more degrees of control freedom. Figure 5 shows the HIRN Meta-Wind Instrument Controller. Signals detected in the instrument mouthpiece include breath pressure, bite tension, lip tension as estimated by measuring the myoelectric activity of the upper lip [Knapp90], and a pitch detector so the player can sing or buzz the lips directly into the instrument mouthpiece. Fingering control is provided via 8 buttons, controlled by four fingers on the right hand, and three fingers and thumb on the left hand. The right hand can be slid linearly along the axis of the instrument, as well as radially rotated. The left hand can be rotated radially. Finally, the head of the instrument can be rotated to vary the playing style from end blown as in a clarinet or soprano saxophone, to edge blown as in a flute. Optional continuous and switch foot controllers can be added to the HIRN controls.

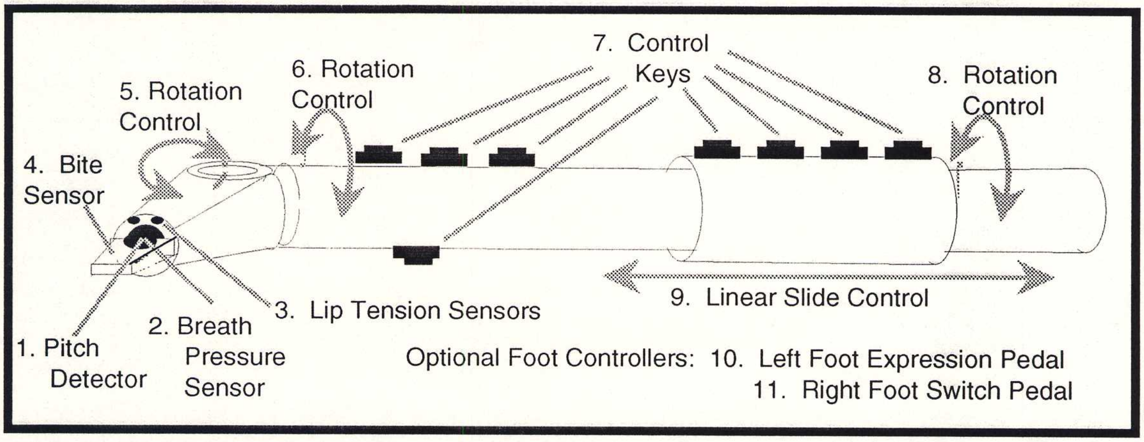


Figure 5. The HIRN meta-wind instrument controller.

4. SUGGESTED MAPPINGS OF HIRN CONTROLS FOR STANDARD MIDI AND WHIRLWIND CONTROL

	Control type	Standard MIDI Control Function	WhirlWind Control Function
1.	Mouthpiece Pitch Detector:	Note # with pitch bend	N/A
2.	Breath Pressure Sensor:	Aftertouch / Breath Controller	Breath Pressure
3.	Lip Tension Sensor:	Note # (with 7.) (Brass Mode)	Lip Tension
4.	Bite Pressure Sensor:	Note # (with 7.) (Wood Mode)	Reed/Lip polynomial
5.	Head Rotation Control:	Cross Fading MIDI Volume Chs.1<-->2 and 3<-->4	Feed-Forward Delay Line (Clar. to Flute)
6.	Left Hand Rotation Control:	Cross Fading MIDI Volume Chs.1&2<-->3&4	Flute Embouchure Control
7.	Control Keys:	Note # Select: Penny Whistle Fingering (Wood) Three Keys as Valves (Brass)	Two implemented as Register Holes
8.	Linear Slide Control:	Pitch Bend	Delay Line Length
9.	Right Hand Rotation Control:	Modulation Wheel	Noise Volume
10.	Left Foot Expression Pedal:	Global Volume Chs. 1, 2, 3, & 4	Output Volume
11.	Right Foot Switch Pedal:	Sustain	Sustain (Breath Pressure)

5. INTERPOLATIONS IN TIMBRAL AND PHYSICAL MODEL SPACE

For MIDI control, the left hand and head rotation controls allow two degrees of movement, controlling a total of four MIDI channel volumes. This simple control scheme has proven quite effective for elementary timbral manipulation. Adding the right hand rotation control would allow for arbitrary movement through a three-dimensional timbral space like that proposed by Grey [Grey75].

A more difficult, but more motivating task is mapping the controls of the HIRN controller to the control parameters of a physical synthesis algorithm like the WhirlWind instrument model. In this case, some of the mappings, like breath pressure and linear slide control, are obvious. The mapping of the head rotation control to the feed-forward delay line gain allows the model to vary quite smoothly from flute-like to clarinet-like. Control of the non-linear polynomial coefficients using linear and Lagrange interpolations has been implemented, and higher-order schemes like those proposed by Bowler et.al. [Bowler90], or neural net schemes like those proposed by Lee and Wessel [Lee92] [Wessel91] could be used to provide smooth interpolation of model parameters.

6. REFERENCES

Bowler, I., P. Manning, A. Purvis, and N. Bailey 1990, "On Mapping N Articulation Onto M Synthesizer-Control Parameters," Proceedings of the ICMC, Glasgow, 181-184.

Cook,P. R. 1991 “TBone: An Interactive WaveGuide Brass Instrument Synthesis Workbench for the NeXT Machine,” Proceedings of the ICMC, Montreal, 297 -299.

Fletcher, N. H. and T. D. Rossing 1991, The Physics of Musical Instruments, New York, Springer Verlag.

Grey J. M. 1975, "An Exploration of Musical Timbre," PhD dissertation, Stanford Department of Music.

Karjalain, M., U. Laine, T. Laakso, V. Valimaki 1991, "Transmission-Line Modeling and Real-Time Synthesis of String and Woodwind Instruments," Proceedings of the ICMC, Montreal, 293 -296.

Knapp, R.B. and H. S. Lusted 1990, "Bioelectric Controller for Computer Music Applications," Computer Music Journal, 14: 1, 42-47.

Lee, M. and D. Wessel 1992, "Connectionist Models for Control of Sound Synthesis," elsewhere in these proceedings.

Smith, J. O. 1987, “Musical Applications of Digital Waveguides." Stanford University Center For Computer Research in Music and Acoustics, Report No. STAN-M-39.

Wessel, D. 1991,"Instruments That Learn, Refined Controllers, and Source Model Loudspeakers," Computer Music Journal, 15: 4, 82-85.

Dynamical Modeling of the Grand Piano Action

Brent Gillespie

Center for Computer Research in Music and Acoustics (CCRMA)
Dept. of Mechanical Engineering, Stanford University, Stanford, CA 94305
email: brent@ccrma.stanford.edu

ABSTRACT

The grand piano action is modeled as a set of four rigid bodies using Kane's method. Computerized symbol manipulation is utilized to streamline the formulation of the equations of motion so that several models can be considered, each of increasing detail. Various methods for checking the dynamical model thus derived are explored. A computer animation driven by simulation of the equations of motion is compared to a high-speed video recording of the piano action moving under a known force at the key. For quantitative evaluation, the velocities and angular velocities of each of the bodies are extracted from the video recording by means of digitization techniques. The aspects of the model of particular interest for emulation by a controlled system, namely, the mechanical impedance at the key and the velocity with which the hammer strikes the string, can be studied in the equations of motion and compared to empirical data.

1. INTRODUCTION

Pianos are judged not only on the basis of their tone, but also feel or 'touch'. The 'dynamics' of the multi-body piano action determine the 'touch' or force history which one feels at the keyboard in response to a given gesture input. Behind these 'dynamics' lie the mechanical properties of the piano action, which are governed by the principles of newtonian mechanics. With tools from the field of applied mechanics, we can build dynamical models whose behaviors approximate those of the piano action. Use of good engineering approximations can be expected to lead to models which are not overly complex, yet descriptive enough to capture the salient properties.

Once such a dynamical model has been devised and proven, a few interesting applications are possible. First, the model allows the testing of piano action (or similar keyboard) designs without having to build prototypes. Secondly, piano simulators can be created. The possibility of synthesis by electronic instruments of not just the sound but also the touch-response of a piano is now within reach. Touch response can be emulated by a keyboard which in fact lacks a piano action but has instead actuators or programmable passive devices and an accompanying control system. A few designs have already been prototyped [Cadoz 1990, Baker 1988].

Analytical investigations into the dynamics of the piano action do exist in the literature [Pfeiffer 1967]. Excellent empirical investigations with a view to characterizing the touch response have also appeared [Askenfelt 1991] However, for the purposes of creating a simulator, an explicit model is desired. This paper documents the development of a dynamical model which fully reflects the effects of inertia properties and changing kinematic constraints on the mechanical impedance for one stroke of a key from rest to hammer/string contact. The various sets of equations of motion which govern this behavior are formulated using Kane's method. Simulations are run and preliminary empirical data taken in order to verify the model.

The remainder of this paper begins with a brief discussion of the piano action behavior as it pertains to touch response. Next, a dynamical model is described and used to formulate the equations of motion. Simulation results are presented and finally, experiments designed to verify the model are discussed.

2. MOTION DESCRIPTION

Touch response can be described in terms of the mechanical impedance (frequency generalized resistance to force) which the piano presents to the player. Mechanical impedance is a function of the inertia, damping, and compliance properties, and also the geometry and interconnection of each of the piano action elements. In this paper, only the behavior resulting from motion beginning at key rest to hammer/string contact will be considered. The functions of the repetition lever and damper are left out. Even this brief motion must be broken into three distinct phases for the purposes of modeling because three sets of kinematic constraints operate. During the first phase, which we will call acceleration, the jack has not yet risen to contact the regulation button (see Figure 1). Therefore, the jack does not rotate with respect to the whippen. The second phase, called let-off, reigns while there is contact between the jack and regulation button, and ends when the interaction forces between the jack and hammer at the knuckle are no longer compressive. The third phase is characterized by free flight of the hammer and further motion of the key, whippen and jack until the key hits the key rail. A model capable of describing the effects of these changing kinematic constraints is desired so that the trigger-like feel of the let-off resistance can be rendered.

3. MODEL DESCRIPTION

Figure 2 shows a schematic representation of the grand piano action to be used for reference in the following discussion. Four bodies comprise the model: The key **K**, the whippen **W**, the jack **J**, and the hammer **H**. (See Figure 2a.) In the present analysis, the function of the repetition lever and the damper lever are not taken into account because they do not play a role during the motion being considered. Let **N** denote a newtonian reference frame. Because only planar motion needs be considered, a point is sufficient to determine an axis of rotation. (See Figure 2c.) Body **K** can rotate in **N** about point $\mathbf{P_1}$ fixed in both **N** and **K**. Similarly, body **W** can rotate in **N** about point $\mathbf{P_2}$ fixed in **N** and **W**. Body **H** can rotate in **N** about point $\mathbf{P_4}$ fixed in **N** and **H** . Finally, body **J** is connected to **W** in such a way that it can rotate about point $\mathbf{P_3}$ fixed in **J** and **W**. Points **H*** and **K*** are the mass centers of bodies **H** and **K**. Moments of inertia for **H** and **K** are also included. Parameters $L_1,..L_{20}$ (not shown) designate pertinent dimensions. To characterize the instantaneous configuration of the action, generalized coordinates $q_1,...q_7$ are employed. (See Figure 2b.) The radian measures of four angles q_1, q_2, q_4, and q_6 are used to locate each of **K**, **W**, **H**, and **J** with respect to the horizontal. Displacement q_3 locates a frictionless slider $\mathbf{S_1}$ which connects **K** to **W**. Displacement q_5 locates a frictionless slider $\mathbf{S_2}$ which connects **J** to **H**. Displacement q_7 locates a frictionless slider $\mathbf{S_3}$ which connects **J** to a horizontal line **RB** fixed in **N** corresponding to the regulation button. All of the sliders $\mathbf{S_1}$, $\mathbf{S_2}$, and $\mathbf{S_3}$ are existent or active, however, only during one phase of the motion of the action: during let-off. Table 1 shows the three phases of motion and the action of each of the sliders to further clarify the manner in which the kinematic constraints evolve.

Readers familiar with the design of the piano action will already be aware of several assumptions made in the above model construction. First, each action element has been modeled as a rigid body. Inertial and weight effects are accounted for, but damping and compliance effects are

Motion Phase	Sub-model Name	Active Sliders	Degrees of Freedon	Number of closed kinematic chains
I	Acceleration	S1 S2 x	1	2
II	Let-off	S1 S2 S3	1	3
IIIa	Hammer Flight	x x x	1	0
IIIb	Key Flight	S1 x S3	1	2

Table 1.

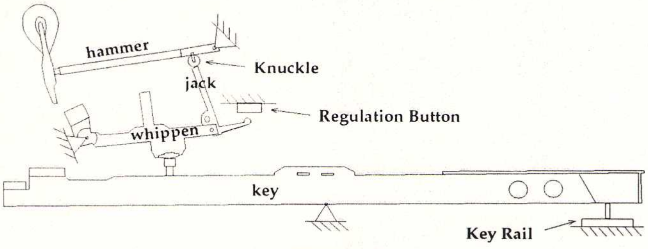


Figure 1.

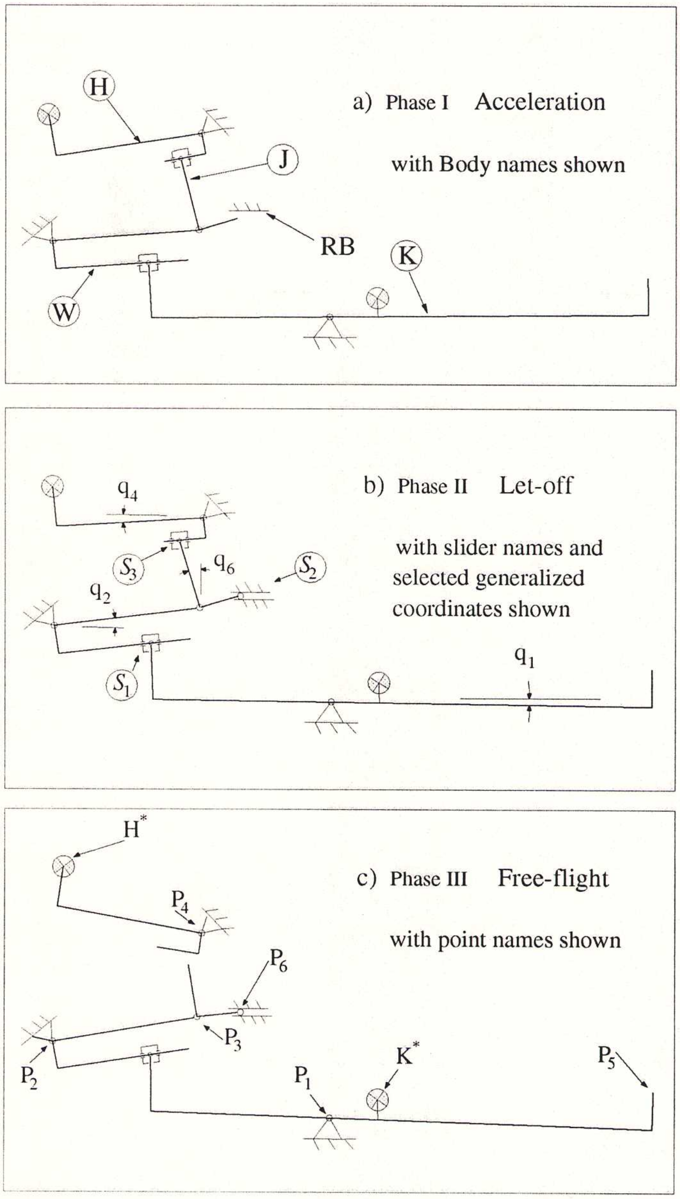


Figure 2.
Schematic diagram of the grand piano action in three configurations corresponding to the three phases of movement: a) acceleration, b) let-off, c) free-flight. The various symbols shown are generally applicable, but are separated over the three configurations for clarity.

not. Rotary friction forces acting at the revolute joints (pin and felt bushings) are neglected. Friction at each of the sliders is also neglected. The above assumptions seem reasonable except perhaps in the case of friction forces which develop during let-off between the jack and hammer at the knuckle. These forces contribute significantly to the 'letoff resistance' felt by a player. Other models not presented here do include these friction effects. Finally, the shape of the contact surfaces between the key and whippen, and jack and hammer are only approximately linear.

4. EQUATION FORMULATION and SIMULATION

Generalized speeds u_i (i=1,...7) are formed as a function of the generalized coordinates simply by letting $u_i = {}^{d}/_{dt}(q_i)$ (i=1,...7). Kinematic constraint equations are used to express u_i (i=2,...7) in terms of u_1. Expressions are found for the velocities of **K***, **H***, and $\mathbf{P_5}$, the point of application of a playing force on the key **K**. The partial velocities for each of these points are found, and are used together with the gravitational forces acting at **K*** and **H*** and the applied force at $\mathbf{P_5}$ to form the generalized active force F_1. Expressions for the accelerations of **K***, and **H*** are found and used with the partial velocities to form the generalized inertia force F_1*. Finally, the dynamical equation of motion is formulated: $F_1+F_1*=0$ [Kane 1985]. An auxiliary generalized speed u_8 is used in Model II to bring the interaction force between the jack and hammer into evidence. In order to determine the reaction forces to a specified velocity input at the key (the inverse-dynamics problem,) each of the models may be formulated as a zero-degree of freedom problem with controlled q_1 time-history. In this case, an auxiliary generalized speed must be used to bring the interaction force between the player and the key into evidence.

The equations of motion, the constraint equations, and expressions for the interaction forces are integrated numerically using a Kutta-Merson algorithm. Care must be taken to ensure that the initial conditions do indeed satisfy the applicable constraint equations. This is accomplished by solving the set of non-linear non-differential constraint equations numerically. A complete model simulation is in hand when simulations of the three sub-models are linked. When point $\mathbf{P_6}$ of body **J** rises to meet line **RB** (tested by a logical statement during simulation,) integration of sub-model I is stopped, and the final conditions are passed as the initial conditions to sub-model II. From then on, slider $\mathbf{S_2}$ constrains $\mathbf{P_6}$ to move along horizontal line **RB**. When the interaction force between **J** and **H** at slider $\mathbf{S_3}$ changes in sub-model II from compressive to tensile (again tested by a logical statement,) integration of sub-model II is stopped and the final conditions are passed as initial conditions to sub-models IIIa and IIIb. At this time, slider S3 disappears leaving two independent models.

A typical equation of motion involves (for each time step) approximately 400 each multiply and add operations and the evaluation of approximately 200 transcendental functions. Simulation in real time on a PC is not possible. However, computerized linearization and other proven simplification techniques can be exploited to speed up the integration process. Simulation output data for the generalized coordinates have been used to place drawings of the four bodies into successive frames which can be compiled into animations. Stick-figure animations (See Figure 3) and full 3-dimensional AutoCad animations have been produced. If the simulation does run in real time, the addition of an actuator whose torque command is driven synchronously by the appropriate generalized coordinate and the linking of simulation input variables to sensors on that actuator constitutes the creation of a simulator. Such schemes have been used to develop virtual piano actions and other virtual dynamical systems [Gillespie 1992].

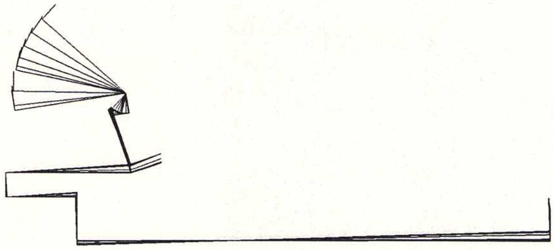


Figure 3
Sample stick-figure animation driven by simulation

5. EXPERIMENT DESCRIPTION

A one-key grand piano action was set into motion by a large linear motor coupled at the key (See Figure 4). Horizontal motions of the linear motor were coupled to vertical motions of the key by means of three bearing pulleys and a loop of Kevlar fiber. A strain gage-instrumented plastic flexure was fastened between the driving Kevlar fibers and the piano key to transduce and record the interaction forces. The linear motor was driven with a parabolic position trajectory identical to the trajectory used as the controlled variable during the inverse-dynamics simulation. Experiments in which a constant force was applied by releasing a weight were also performed. Position encoders recorded the resulting motions of the key, Kevlar loop, and damper. A high-speed video recording was made at 1000 frames per second of the motion of the piano action. Retro-reflective patches were attached to ten locations on the piano action in order to facilitate vision recognition by computer. Digitization and light patch centroid location determination from about 700 frames was performed by Jim Walton of 4-D Video. The digitized motions were used to deduce the generalized coordinate trajectories with direction cosine matrix transformations. The theoretical (simulated) and experimental generalized coordinate trajectories can be compared. Some preliminary results are presented in Figure 5. Such comparisons are serving to improve the model and further direct experiment design.

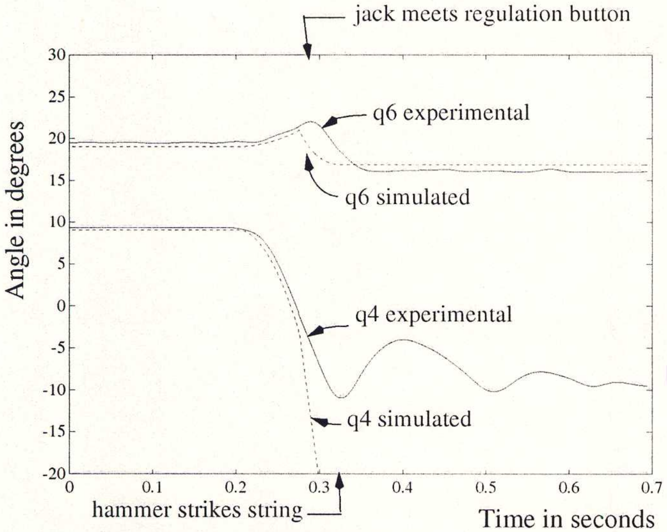


Figure 4

Generalized coordinates q4 and q6 vs. time experimental and simulated.

6. SUMMARY

The use of modern dynamical modeling techniques allows the development of accurate and descriptive models of the grand piano action mechanical impedance. This paper has documented a dynamical model which fully reflects the effects of inertia properties and changing kinematic constraints on the mechanical impedance for one stroke of a key from rest to hammer/string contact. This model captures the major characteristics of the piano action behavior, as verified by preliminary experiments. Such models can serve as the cornerstone in a haptic display device design. Model enhancements are the subject of continuing research at CCRMA.

ACKNOWLEDGEMENTS

I wish to thank Dr. Jim Walton of 4D Video, Sebastopol, CA, professor Tom Bowman for the loan of the SP2000 high speed video camera, Yamaha Corporation for donation of a one-key piano action and Delta Tau corporation for donation of a DSP motor control card.

REFERENCES

A. Askenfelt and E. V. Jansson, ™From touch to string vibration II: The motion of the key and hammer,
J. Acoust. Soc. Am, vol. 90 (5), pp. 2383-2393, Nov. 1991.

R. Baker, ™Active touch keyboard, United States Patent No. 4,899,631, 1988..

C. Cadoz, L. Lisowski, and J-L. Florens, ™Modular feedback keyboard, Proceedings of the ICMC, pp. 379-382, Glasgow, 1990.

B. Gillespie, ™The touchback keyboard, in these proceedings, 1992.

T. R. Kane, and D. A. Levinson, Dynamics: Theory and Applications, McGraw-Hill, New York, 1985, P. 158.

W. Pfeiffer, The Piano Key and Whippen, Verlag Das Musikinstrument, Frankfurt a. M, 1967.

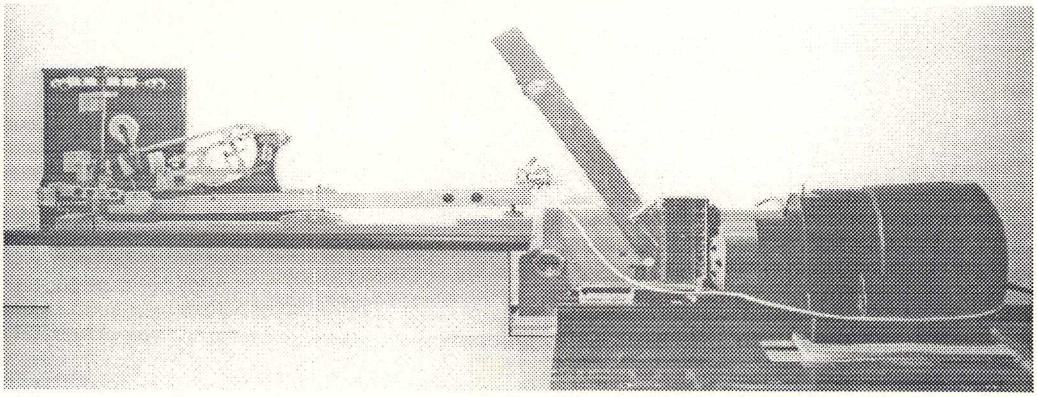


Figure 4.
Experimnetal Setup

The Touchback Keyboard

Brent Gillespie

Center for Computer Research in Music and Acoustics (CCRMA)
Dept. of Mechanical Engineering, Stanford University, Stanford, CA 94305
email: brent@ccrma.stanford.edu

ABSTRACT

A digital control system capable of simulating multi-degree-of-freedom dynamical systems in real time with visual, audio, and haptic displays is presented. A set of software and hardware tools form a testbed in which a dynamical system can be modeled, reduced to equations of motion, and simulated. The user interacts with a powered key to influence the behavior of the dynamical system and feel the computed interaction forces being fed back in real-time. Of primary interest for simulation with haptic display are the grand piano action and other keyboard instrument controllers. Various keyboard actions are demonstrated.

1. INTRODUCTION

Synthesis of not only the sound, but also the touch response of a musical instrument is made possible when a synthesizer interface contains actuators and control systems. Active (or even just programmable passive) components take the place of the 'mechanics' of the acoustic instrument (such as a piano action) while preserving the dynamical response characteristics or 'feel'. This design challenge has been taken on by researchers at several institutions, including ACROE [Cadoz 1990] and CCRMA, and independent inventors [Baker 1988]. A keyboard controller of this type can make various touch responses, such as those of a harpsichord, organ, or grand piano, available at the touch of a button. The advantages, however, go even beyond making synthesizers 'feel right'; the broader goal is to re-establish the touch relationship between performer and instrument.

How do we know, when we play an instrument, what effect our manipulations are having? Certainly we hear the response of the instrument; information flows from instrument to performer through sound. But with which additional senses do we follow the responses of an instrument to our gestures? The haptic senses: tactile, kinesthetic, force, proprioperceptive , etc. Unlike the audio, this mechanical channel allows bi-directional information flow. Since the only manner for the performer to communicate to the instrument is through manipulation by hand or mouth, these force/motion trajectories must transmit the performer's intentions to the instrument. Conversely, the reaction force/response motion history with which the instrument answers the input gesture is a signal containing valuable information about the instrument's behavior.

When, in addition to instantaneous force/velocity data, a force/velocity history is available, as is the case in the response to a gesture, the performer can perceive whether he/she is acting through an inertia, damper, compliance, or combination of these to excite sound vibrations. We humans, equipped both with a means of manipulation and with haptic senses, are ideally suited to explore the 'physics' of our environments. In fact, processing feel information with the brain and using it to modify manipulation may be faster than processing and responding to audio information. [Phillips 1987]. We use such terms as 'look and feel' to refer to the response behavior of an interface to our input manipulations. An interface with a 'good feel' conveys maximum, even redundant information about the state of an application or whatever lies behind the interface. This allows a user to manipulate efficiently toward a given goal. Aftertouch on synthesizer keyboards is an example of an interface with no significant 'feel' feedback. There is no opportunity for the user to sense the state of the system except by listening. By contrast, a correspondence (not necessarily one-to-one) does exist between the touch-response and the sound of acoustic instruments. Although the haptic information may be redundant, it plays a vital role in processes such as learning to elicit desired tones.

The remainder of this paper introduces engineering language into the above discussion, then uses these terms to describe specifics about the touch-response of the grand piano and to outline specific design goals for a touch-programmable keyboard. Finally, a touchback keyboard prototype, useful for exploring these issues, is described.

2. MECHANICAL IMPEDANCE

The touch-programmable keyboard design challenge is similar in scope and make-up to that faced by designers of telerobots with force-reflection and haptic display devices for virtual environments. Motorized manipulanda relay a range of mechanical information either from the physical environment in which the slave manipulator operates or from the simulated virtual environment. There is a rich and growing body of literature in the field of robotics which can be profitably drawn upon. [Millman and Colgate 1991]

The performer and instrument are dynamical systems made up of inertial, damping and compliant elements, and, in the case of the performer, active elements. These two systems may exchange energy through an interaction port when contact is made between them. An equivalent passive impedance (frequency generalized resistance to force) may be substitued for the instrument. This manner of decomposition is completely analogous to techniques commonly used in circuit analysis. Whatever mechanism might lie behind a key, its influence on the 'feel' can be replaced by an instantaneous effective inertia, effective damping, and effective compliance at the key. The instantaneous reaction force that one feels is completely specified by three proportionality constants, one each for acceleration, velocity, and position. The only extension needed (in order to cover cases in which contacts are made and broken between various members of the mechanism) is to allow these three parameters to take on a configuration dependence. That is, because various sets of kinematic constraints are operative at various positions of a piano key, various inertial, damping or compliant elements will be active, or their effects reflected through various sets of lever arms.

3. THE FEEL OF THE GRAND PIANO

The following types of behavior characterize the piano feel and serve as targets for the design of a touch-programmable keyboard. First, the impedance of the piano action is dominated by the inertia of the hammer because of the catapult-like function of the action. There is an approximately 5-times mechanical advantage of the key over the hammer. Along with this inertia force, the performer feels a constant return force due to gravity acting on the hammer and key, called the 'static imbalance'. When the instrument is played slowly, a dissipative force becomes apparent just before key-bottom, called 'let-off resistance.' This behavior results from friction between the jack and hammer knuckle as they slide against one-another during let-off. Finally, it is desired to re-create the repetition capabilities of the grand piano in the touch-programmable keyboard. The performer should be able to bounce a virtual hammer and feel the set, and then the re-trigger function of the repetition lever and jack at work.

4. DESIGN GOALS

The physical portion of the interface, the key and connecting actuators, are subject to the following design guidelines. To provide the performer with maximum sensitivity to variation in the level of computed force, the physical device itself must exhibit low inertia. To avoid increasing force thresholds and degrading force resolution, it must have low friction. To avoid adding unwanted compliant-behavior dynamics and to side-step potential control instability problems, the device, including its drive train, should exhibit high stiffness. The device must be highly back-driveable and have little or no backlash. The range of forces should be matched to human capabilities if the keyboard is to serve as a general experimental device, or, if it is to emulate existing instruments such as the piano, be matched to these. Whether the keyboard should be capable of simulating hard surfaces by active means or by passivė means should be carefully considered because such simulations place very high demands on the maximum available force, the servo rate, and the controller robustness. The range of motion can be physically limited yet, in some sense, be made programmable by industrious design. Finally, a most exacting requirement arises with regard to size and weight. The physical device must be portable and wieldy. To be added to the list of design goals before this device hits the market, is cost.

5. THE TOUCHBACK KEYBOARD DESIGN

The present prototype meets only a subset of the above design goals, but does provide a useful testbed for exploring various control algorithms. A keyboard of eight keys has been constructed. Each key is coupled stiffly to its own voice-coil type linear motor, originally designed for use in large disk drives. Because these motors are somewhat oversized and of rather high inertia, one version of the interface is scaled up in size to a carillon keyboard to be played with the hands instead of a piano keyboard to be played with the fingers. Each motor is in turn driven by its own independent voltage controlled current amplifier. A 40386-based PC plays host to an eight-channel DSP motor control card containing all the necessary D/A, A/D and de-code hardware. The keys themselves are equipped with optical encoders, strain gages, and tachometers to sense position, velocity and performer/key interaction force. Knowing both the performer/key interaction force and the force applied by the motor to the key, the acceleration of the key can be deduced, providing a more accurate estimation of the actual acceleration than measurement with an accelerometer or differentiation of the velocity signal. The control loop can be closed either in the DSP chip itself, if speed is needed, or in a C program on the '386 cpu where the logic and control scheme design are very accessible. Various control schemes are being explored. For example, a description of the piano action in functional control blocks and logic can become the basis of the control system architecture. Another approach involves first formulating the equations of motion from the mechanical description and then writing these into a controller based on a numerical integration scheme [Gillespie 1992].

6. CLOSING

A few of the analysis and design techniques which have proven useful in the field of robotics are being directed toward the touch-programmable keyboard design problem. If the synthesized sound is actively evolving, as is often the case on popular new patches, performer instrument interaction which involves energy exchanges in both directions offers exciting new musical composition and performance possibilities.

REFERENCES

R. Baker, "Active Touch Keyboard," United States Patent No. 4,899,631., 1988.

C. Cadoz, L. Lisowski, and J-L. Florens, "Modular Feedback Keyboard," *Proceedings of the ICMC*, pp. 379-382, Glasgow, 1990.

B. Gillespie, "Dynamical modeling of the grand piano action," in these proceedings, 1992.

P. A. Millman and J. E. Colgate, "Design of a four degree-of-freedom force-reflecting manipulandum with a specified force/torque workspace," *Proceedings of the IEEE Int. Conference on Robotics and Automation*, vol. II pp. 1488-1493, April, 1991.

C. G. Phillips, "Brains and Hands," Music and the Brain, M. Critchley and R. A. Henson, eds., pp. 48-58, Charles Thomas, Springfield, IL 1977.

The Second-Order Digital Waveguide Oscillator

Julius O. Smith and Perry R. Cook

Center for Computer Research in Music and Acoustics (CCRMA)
Music Dept., Stanford University, Stanford, CA 94305
email: jos/prc@ccrma.stanford.edu

Abstract

A digital sinusoidal oscillator derived from digital waveguide theory is described which has good properties for VLSI implementation. Its main features are *no wavetable* and a computational complexity of only *one multiply* per sample when amplitude and frequency are constant. Three additions are required per sample. A piecewise exponential amplitude envelope is available for the cost of a second multiplication per sample, which need not be as expensive as the tuning multiply. In the presence of frequency modulation (FM), the amplitude coefficient can be varied to exactly cancel amplitude modulation (AM) caused by changing the frequency of oscillation.

1. Introduction

One of the very first computer music techniques introduced was *additive synthesis* (Risset 1985). It is based on Fourier's theorem which states that any sound can be constructed from elementary sinusoids, such as are approximately produced by carefully struck tuning forks. Additive synthesis attempts to apply this theorem to the synthesis of sound by employing large banks of sinusoidal oscillators, each having independent amplitude and frequency controls. Many analysis methods, e.g., the phase vocoder, have been developed to support additive synthesis. A summary is given in (Serra and Smith 1990).

While additive synthesis is very powerful and general, it has been held back from widespread usage due to its computational expense. For example, on a single DSP56001 digital signal-processing chip, clocked at 33 MHz, only about 60 sinusoidal partials can be synthesized in real time using non-interpolated, table-lookup oscillators. Interpolated table-lookup oscillators are much more expensive, and when all the bells and whistles are added, and system overhead is accounted for, only around 12 fully general, high-quality partials are sustainable at 44.1 KHz on a $33MHz$ DSP56001 (based on analysis of implementations provided by the NeXT Music Kit).

At CD-quality sampling rates, the low A on the piano requires $22050/55 \approx 400$ sinusoidal partials, and at least the low-frequency partials should use interpolated lookups. Assuming a worst-case average of 100 partials per voice, providing 32-voice polyphony requires 3200 partials, or around 64 DSP chips, assuming we can pack an average of 50 partials into each DSP. A more reasonable complement of 8 DSP chips would provide only 4-voice polyphony which is simply not enough for a piano synthesis. However, since DSP chips are getting faster and cheaper, DSP-based additive synthesis looks viable a year or two out.

The cost of additive synthesis can be greatly reduced by making special purpose VLSI optimized for sinusoidal synthesis. In a VLSI environment, major bottlenecks are *wavetables* and *multiplications.* Even if a single sinusoidal wavetable is shared, it must be accessed sequentially, inhibiting parallelism. The wavetable can be eliminated entirely if *recursive algorithms* are used to synthesize sinusoids directly.

In (Gordon and Smith 1985), three techniques were examined for generating sinusoids digitally by means of recursive algorithms. The recursions can be interpreted as implementations of second-order digital resonators in which the damping is set to zero. The three methods considered were (1) the *coupled form* which is identical to a two-dimensional vector rotation, (2) the *modified* *coupled form,* or "magic circle" algorithm, which is similar to (1) but has ideal numerical behavior, and (3) the direct-form, second-order, digital resonator with its poles set to the unit circle. These three recursions are defined as follows.

$$\begin{aligned}
(1)\qquad x_n &= c_n x_{n-1} + s_n y_{n-1}\\
y_n &= -s_n x_{n-1} + c_n y_{n-1}\\
(2)\qquad x_n &= x_{n-1} + \epsilon y_{n-1}\\
y_n &= -\epsilon x_n + y_{n-1}\\
(3)\qquad x_n &= 2c_n x_{n-1} - y_{n-1}\\
y_n &= x_{n-1}
\end{aligned}$$

where $c_n \stackrel{\Delta}{=} \cos(2\pi f_n T)$, $s_n \stackrel{\Delta}{=} \sin(2\pi f_n T)$, f_n is the instantaneous frequency of oscillation (Hz) at time sample n, and T is the sampling period in seconds. The magic circle parameter is $\epsilon = 2\sin(\pi f_n T)$.

The digital waveguide oscillator appears to have the best overall properties yet seen for VLSI implementation. The new structure was derived as a spin-off from recent results in the theory and implementation of digital waveguides (Smith 1987a; Smith 1987b). Any second-order digital filter structure can be used as a starting point for developing a corresponding sinusoidal signal generator, so in this case we begin with the second-order wavguide filter.

2. The Second-Order Waveguide Filter

The first step is to make a second-order digital filter with zero damping by abutting two unit-sample sections of waveguide medium, and terminating on the left and right with perfect reflections, as shown in Fig. 1. It turns out that to obtain sinusoidal oscillation, one of the terminations must provide an inverting reflection while the other is non-inverting.

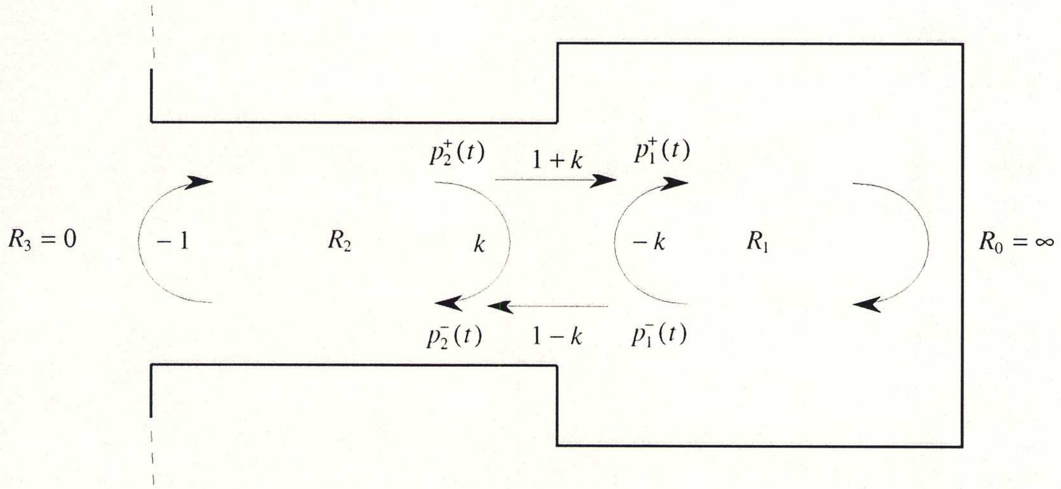


Figure 1. The second-order, lossless, digital waveguide oscillator, built using two *acoustic tube* sections. The wave impedance in section i is given by $R_i = \rho c/A_i$, where ρ is air density, A_i is the cross-sectional area of tube section i, and c is sound speed. The reflection coefficient is determined by the impedance discontinuity via $k = (R_1 - R_2)/(R_1 + R_2)$.

At the junction between sections 1 and 2, the signal is partially transmitted and partially reflected such that energy is conserved, i.e., we have *lossless scattering.* The formula for the reflection coefficient k can be derived from the physical constraints that (1) pressure is continuous across the junction, and (2) there is no net flow into or out of the junction. For traveling pressure waves $p^{\pm}(t)$ and volume-velocity waves $u^{\pm}(t)$, we have $p^{+}(t) = Ru^{+}(t)$ and $p^{-}(t) = -Ru^{-}(t)$. The physical pressure and volume velocity are obtained by summing the traveling-wave components.

The discrete-time simulation for the physical system of Fig. 1 is shown in Fig. 2. The propagation time from the junction to a reflecting termination and back is one sample period. The half sample delay from the junction to the reflecting termination has been *commuted* with the termination and combined with the half sample delay *to* the termination. This is a special case of a "half-rate" waveguide filter (Smith 1987a).

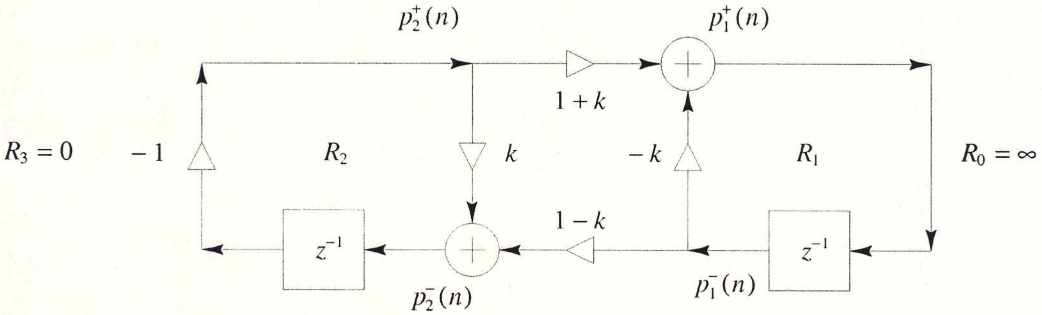


Figure 2. The second-order, lossless, waveguide filter.

Since only two samples of delay are present, the digital system is at most second order, and since the coefficients are real, at most one frequency of oscillation is possible in $(0, \pi)$.

The scattering junction shown in the figure is called the Kelly-Lochbaum junction in the literature on lattice and ladder digital filters (Markle and Gray 1975,1976). While it is the most natural from a physical point of view, it requires four multiplies and two additions for its implementation.

It is well known that lossless scattering junctions can be implemented in a variety of equivalent forms, such as the *two-multiply* and even one-multiply junctions. However, most have the disadvantage of not being *normalized* in the sense that changing the reflection coefficient k changes the amplitude of oscillation. This can be understood physically by noting that a change in k implies a change in R_2/R_1. Since the signal power contained in a waveguide variable, say $p_1^+(n)$, is $\left[p_1^+(n)\right]^2/R_1$, we find that modulating the reflection coefficient corresponds to modulating the signal energy represented by the signal sample in at least one of the two delay elements. Since energy is proportional to amplitude squared, energy modulation implies amplitude modulation.

The well-known normalization procedure is to replace the traveling pressure waves $p^{\pm}$ by "root-power" pressure waves $\tilde{p}^{\pm} = p^{\pm}/\sqrt{R}$ so that signal power is just the square of a signal sample $(\tilde{p}^{\pm})^2$. When this is done, the scattering junction transforms from the Kelly-Lochbaum or one-multiply form into the *normalized ladder* junction in which the reflection coefficients are again $\pm k$, but the forward and reverse transmission coefficients become $\sqrt{1-k^2}$. Defining $k = \sin(\theta)$, the transmission coefficients can be seen as $\cos(\theta)$, and we arrive essentially at the *coupled form,* or two-dimensional vector rotation considered in (Gordon and Smith 1985).

An alternative normalization technique is based on the digital waveguide *transformer* (Smith 1987a). The purpose of a "transformer" is to "step" the force variable (pressure in our example) by some factor g without scattering and without affecting signal energy. Since traveling signal power is proportional to pressure times velocity p^+u^+, it follows that velocity must be stepped by the inverse factor $1/g$ to keep power constant. This is the familiar behavior of transformers for analog electrical circuits: voltage is stepped up by the "turns ratio" and current is stepped down by the reciprocal factor. Now, since $p^+ = Ru^+$, traveling signal power is equal to $p^+u^+ = (p^+)^2/R$. Therefore, stepping up pressure through a transformer by the factor g corresponds to stepping up the wave impedance R by the factor g^2. In other words, the transformer raises pressure and decreases volume velocity by raising the wave impedance (narrowing the acoustic tube) like a converging cone. (Since the transformer steps the impedance instantaneously, and since scattering occurs at impedance discontinuities, the transformer has no precise physical analogue, although it can be approximated using a conic frustum.)

If a transformer is inserted in a waveguide immediately to the left, say, of a scattering junction, it can be used to modulate the the wave impedance "seen" to the left by the junction without having to use root-power waves in the simulation. As a result, the one-multiply junction can be used for the scattering junction, since the junction itself is not normalized. Since the transformer requires two multiplies, a total of three multiplies can effectively implement a normalized junction, where four were needed before. Finally, in just this special case, one of the transformer coefficients can be commuted with the delay element on the left and combined with the other transformer coefficient. For convenience, the -1coefficient on the left is commuted into the junction so it merely toggles the signs of inputs to existing summers. These transformations lead to the final form shown in Fig. 3.

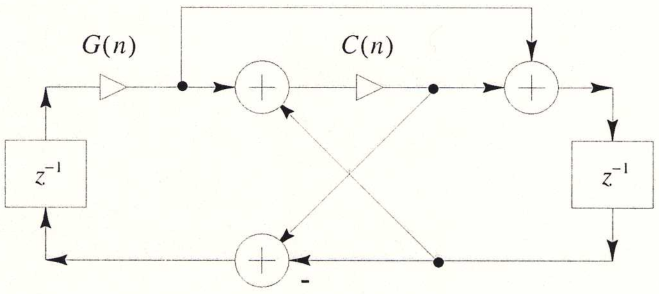


Figure 3. The transformer-normalized, digital waveguide oscillator.

The "tuning coefficient" is given by $C(n) = \cos(2\pi f_n T)$, where f_n is the desired oscillation frequency in Hz at sample n, and T is the sampling period in seconds. The "amplitude coefficient" is $G(n) = r_n g_n / g_{n-1}$, where $r_n = e^{-T/\tau_n}$ is the exponential growth or decay per sample ($r_n \equiv 1$ for constant amplitude), and g_n is the normalizing transformer "turns ratio" given by $g_n = \sqrt{[1 - C(n)]/[1 + C(n)]}$. When both amplitude and frequency are constant, we have $G(n) \equiv 1$, and only the tuning multiply is operational. When frequency changes, the amplitude coefficient deviates from unity for only one time sample to normalize the oscillation amplitude.

When amplitude and frequency are constant, there is no gradual exponential growth or decay due to round-off error. This happens because the only rounding is at the output of the tuning multiply, and all other computations are exact. Therefore, quantization in the tuning coefficient can only cause quantization in the frequency of oscillation. Note that any one-multiply digital oscillator should have this property. In contrast, the only other known normalized oscillator, the coupled form, *does* exhibit exponential amplitude drift because it has *two* coefficients $c = \cos(\theta)$ and $s = \sin(\theta)$ which, after quantization, no longer obey $c^2 + s^2 = 1$ for most tunings.

Conclusions

A recursive algorithm was presented for digital sinusoid generation that has excellent properties for VLSI implementation. It is like the coupled form in that it offers instantaneous amplitude from its state and constant amplitude in the presence of frequency modulation. However, its implementation requires only one or two multiplies per sample instead of four.

While these properties make the new oscillator appear ideally suited for FM applications in VLSI, there are issues to be resolved regarding conversion from modulator output to carrier coefficients. Preliminary experiments indicate that FM indices less than 1 are well behaved when the output of a modulating oscillator simply adds to the coefficient of the carrier oscillator (bypassing the exact FM formulas). Approximate amplitude normalizing coefficients have also been derived which provide a first-order approximation to the exact AM compensation at low cost. For music synthesis applications, we believe a distortion in the details of the FM instantaneous frequency trajectory and a moderate amount of incidental AM can be tolerated since they produce only second-order timbral effects in many situations.

References

J. W. Gordon and J. O. Smith, "A Sine Generation Algorithm for VLSI Applications," *Proc. 1985 Int. Conf. Computer Music,* pp. 165–168.

A. H. Gray and J. D. Markel, "Digital Lattice and Ladder Filter Synthesis," *Selected Papers in Digital Signal Processing,* Digital Signal Processing Committee, eds., IEEE Press, 1975.

J. D. Markel and A. H. Gray, *Linear Prediction of Speech,* Springer-Verlag, New York, 1976.

P. M. Morse and K. U. Ingard, *Theoretical Acoustics,* McGraw-Hill, New York, 1968.

J. C. Risset, "Computer Music Experiments 1964—...," *Computer Music Journal,* 9(1):11–18, Spring, 1985. Reprinted in *The Music Machine,* , Roads, C., ed., MIT Press, 1989.

X. Serra and J. O. Smith, "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition," *Computer Music Journal,* 14(4):12–24, Winter, 1990.

J. O. Smith, "Music Applications of Digital Waveguides," CCRMA Tech. Rep. STAN–M–39, Stanford University, 1987. This is a compendium of four papers and presentations. Copies can be ordered by calling CCRMA at (415)723-4971.

J. O. Smith, "Waveguide Filter Tutorial," *Proc. 1987 Int. Conf. Computer Music,* Illinois.

J. O. Smith, "Unit-Generator Implementation on the NeXT DSP Chip," *Proc. 1989 Int. Conf. Computer Music,* pp. 303–306.

Implementing Quadraphonic Audio on the NeXT:
Hardware & Software Issues

Atau Tanaka

CCRMA
Music Depatment
Stanford University
Stanford, CA 94305-8180 U.S.A.
atau@ccrma.stanford.edu

ABSTRACT

Implementation of extensions to the audio capabilities of the NeXT to include four channel sound output is described. Low level software was written to make the quadraphonic system accessible to the composer from a high level in an object oriented fashion. The system uses the single DSP56000 onboard the NeXT, with the addition of outboard digital/analog convertor hardware. Applications include continuation of previous four channel computer music, real time performance of spatial audio pieces, and sound localization perception research.

INTRODUCTION

Four channel audio playback has, over the years, become a tradition in electronic and computer music production and concert presentation. In the past several years, many in the computer music community have chosen the NeXT computer as their working platform, due to its extensive inherent music capabilities, which include a built in digital signal processor (DSP) and high quality digital to analog convertors (DACs). Despite these features that make it the ideal computer music machine, the NeXT lacks quadraphonic audio outputs. Thus, when the NeXT replaced the Foonly/Samson Box system at CCRMA, the studio lost its quadraphonic music production facilities. The following presents a project undertaken to extend the audio capabilities of the NeXT to include four channel sound output.

GOALS

Four channel audio capability has several applications: 1) continuation of previous computer music work, including Chowning's Quad program for moving sound source simulation on the Foonly (Chowning 1971), 2) real time performance of spatial audio pieces, 3) sound localization perception research. This system is inspired by and extends the wide range of previous work (Bosi 1990, Kendall 1981).

The primary musical goal is the creation of a system that gives composers a working environment to write music with spatial sound, and that provides a concert playback system for such music. Existing systems require the composer to access special purpose hardware and studio environments to create music with spatial qualities. Much focus has been placed on headphone based and two channel speaker systems. Headphone systems are not applicable for concert playback. Although two channel speaker systems are advancing, they are still not entirely effective for all listening positions in a large concert space.

A four channel system can be effective due to its simplicity. With the proposed system, a composer will need only a Motorola DSP56001 equipped computer, a low cost DAC, and a four channel amplifier/speaker system - all general purpose items that are available for a modest cost. Likewise for concert playback, a four channel sound reinforcement system can be readily set up in almost any hall. The physical existence of sound sources in all four quadrants of the horizontal Cartesian plane reduces the computational overhead required to overcome perceptual phenomena such as front/rear confusion (Blauert 1983).

The approach of two speaker systems has been to attempt to simulate an anechoic binaural listening environment in a space that is inherently not. Such an approach is prone to "sweet spots," where the system is effective only for a few ideal listening positions. A four channel system alleviates this need and accommodates a larger prime listening area. Such qualities are inherent to the four channel speaker system alone. This frees up computing resources to be allocated to processes that make the best use of a space rather than try to compensate for the existence of one. For example, with front/back disparities taken care of by the speaker system, signal processing resources can be applied to height and distance simulation.

IMPLEMENTATION

Implementation goals include a consistent and transparent extension of existing NeXT music/audio facilities. In software, this means making the system accessible from a high level in an object oriented fashion. In hardware, it means using the single DSP56001 on board. To realize such a system requires low level extensions in software, and the addition of outboard output hardware.

The initial system made use of the built in DACs with an additional pair of output channels. The audio for the auxiliary channels was sent out the DSP's serial port to outboard stereo DACs. Software was written that read a four channel sound file from disk (in NeXT's standard sound file format), and sent each channel to its respective output channel. Writing to the built in DACs and to the DSP serial port (SSI) required the use of two different protocols. The DSP is able to play sound out of the internal DACs via a direct memory access (DMA) stream. This takes place in units of a defined DMA buffer. Since the SSI is a peripheral on chip to the DSP, no DMA is needed. A double buffering scheme is set up in the registers of the DSP, and data is transmitted in an interrupt driven manner.

With what seemed like the simplest scenario of two internal DAC channels and two outboard DAC channels, software issues arose that jeopardized the simplicity. The DMA protocol and serial interrupt handling mechanisms had to be reconciled and synchronized. Because the DSP is not able to handle the two processes concurrently, synchronization error could be no better than one DMA buffer's worth of time. With a 512 sample buffer at a 44.1KHz sampling rate, this error was in excess of 10ms. This approach, then, was not accurate enough for musical applications, and introduced unneeded complexity in software.

Given this, it was decided to create a device that would connect to the DSP serial port and provide all four channels of output. Philips/Signetics donated several of their recent stereo digital filter/DAC chips, often found in CD player applications. This gave us the opportunity to create a new low cost four channel DAC by using two such chips in parallel and taking advantage of their alternating data/zero data format. At present, a two channel prototype has been completed. It has been shown to work by receiving data from a Crystal Semiconductor analog/digital convertor (ADC). Connection to the NeXT requires a data format issue to be resolved. The Philips DAC expects data input in the form of Philips' I^2S format. This format specifies that the data word begins one bit after the rising edge of a frame sync signal. The NeXT software used to send data out the SSI currently sends the first data word at the same time as the rising edge of the frame sync. A flip flop circuit is being implemented to reconcile this format difference.

The use of a single output device for all four audio channels simplifies the DSP software considerably. This will free up resources on the DSP56001 for additional tasks such as the implementation of Doppler shift and panning algorithms.
The development of these quadraphonic hardware/software extensions to the NeXT marks only the beginning of more interesting spatial sound work to come. With this low level support scaffold in place, the user will be able to continue working at a high level, whether it be in Objective C or LISP. Four channel output capability will simply be another high level call. These extensions will serve as valuable research and compositional tools and will provide a real time multichannel concert playback engine.

ACKNOWLEDGMENTS

This research was made possible through donations from Crystal Semiconductor, Motorola, and Philips/Signetics. The author wishes to thank Jay Kadis, Fernando Lopez Lezcano, Stephen Davis, and Jimmi Hagen for their invaluable support and assistance.

REFERENCES

M. Bosi, "An Interactive Real-time System for the Control Sound Localization," *Computer Music Journal*, v.14n.4, pp. 59-64, Winter 1990.

J. M. Chowning, "The Simulation of Moving Sound Sources," *Journal of the Audio Engineering Society*, v.19, pp. 2-6, 1991.

G. S. Kendall and C. A. P. Rodgers, "The Simulation of Three-Dimensional Localization Cues for Headphone Listening," *Proceedings of the 1981 International Computer Music Conference*, pp. 225-243, 1981.

J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, MA, 1983.

Implementation of a Variable Pick-Up Point on a Waveguide String Model with FM/AM Applications

Scott A. Van Duyne and Julius O. Smith, III

Center for Computer Research in Music and Acoustics (CCRMA)
Music Dept., Stanford University, Stanford, CA 94305
email: savd/jos@ccrma.stanford.edu

Abstract

The waveguide string model can be extended through the addition of a movable pick-up point along the string. As the pick-up slides along the string, a flanging effect results. Accelerating the pick-up produces glissandi effects. Modulating the placement of the pick-up sinusoidally at audio rates produces modified FM sidebands around each partial on the string, with control over the presence of even and odd sidebands. This non-physical effect can be used to enrich the timbre of physical models, taking advantage of the well-understood theory of FM synthesis.

1. Background and Description of the Model

The bidirectional traveling displacement waves on an ideal string can be represented by a waveguide synthesis model. The actual displacement of the string at any point is determined by adding the displacements associated with the left- and right-going traveling waves in the bidirectional delay lines. The ideal reflection of the traveling waves at each end of the string is modeled by multiplying the incoming wave by -1 and sending it through in the other direction. Figure 1 illustrates the ideal string model.

To listen to the vibrations on this string, a pick-up must be placed at some point on the string. For our purposes, we would like to place a pick-up at any continuous point along the string. This necessitates an interpolation of some kind. If the pick-up is to be placed exactly at a sample point, the values of the upper and lower delay lines at that point are simply added. If the pick-up is to be placed at a fractional distance, α, between two sample points, then a weighted average of adjacent sample values must be performed. In this initial research, linear interpolation is used. The results are excellent, but there is room for a better interpolation method in extreme cases of pick-up point motion. Figure 2 shows a close-up of the implementation of an interpolated pick-up point.

Figure 1.

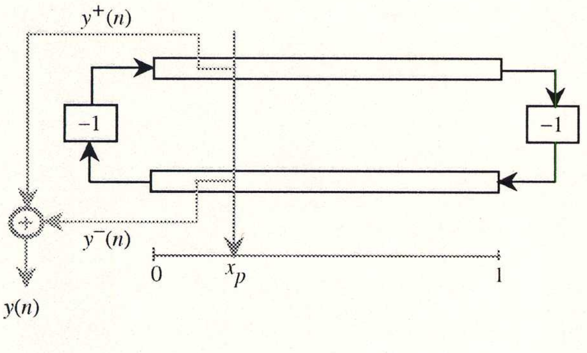
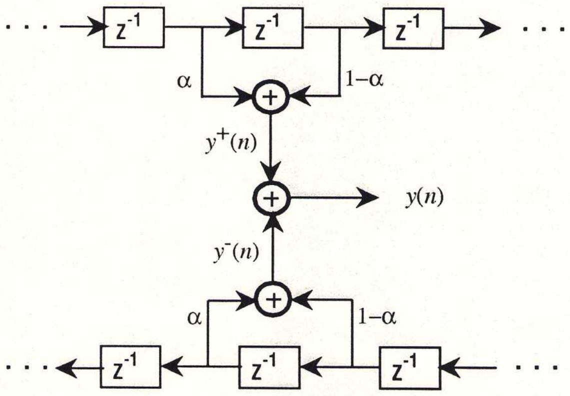


Figure 2.



2. Moving the Pick-Up Point

Acoustically, the position of a pick-up on a string determines the relative magnitudes of the partials heard. If it is placed at one of the end points, no sound will be present since the end points do not move. If the pick-up is placed in the exact center of the string, only odd harmonics will be present in the spectrum because all even harmonics will have a node at that point. In general, placing the pick-up at a position 1/k across the string will zero out partials with harmonic number a multiple of k, and reduce levels of near multiples of k. The placement of a pick-up is therefore essentially a comb filter.

Sliding the pickup along the string slowly creates a flanging effect as the nodes of the various harmonics are crossed. Figure 3 shows a sonogram view of the spectrum of a sliding pick-up point. In this picture, the time axis is horizontal, and the frequency axis is vertical. The magnitude is indicated by relative lightness and darkness. Here, the pick-up slides across the length of the string at constant speed over the duration of the sound. The evolution of the first twelve harmonics on the string are shown in this sonogram view.

Figure 3.

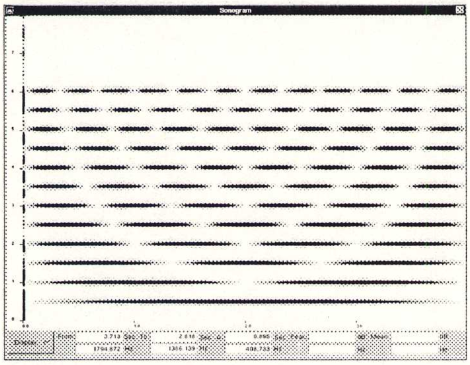
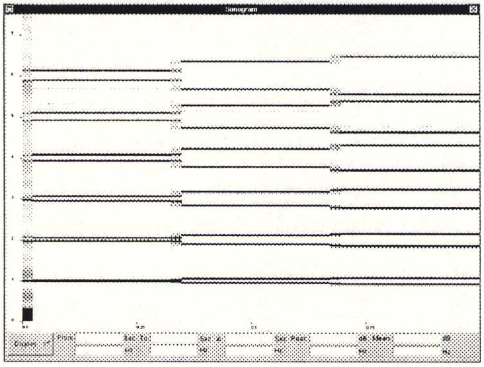


Figure 4.



When we slide the pick-up along the string, we are, in effect, catching up with one of the waves and losing ground with the other. This results in a separation of their frequencies due to Doppler shift in opposite directions. In an alternative view, since sliding the pick-up across the string at a constant rate modulates the amplitude of each harmonic on the string sinusoidally at a frequency of $0.5\ k\ s$, where $0.5\ k$ is the number of periods of the k^{th} harmonic standing on the string, and s is the number of string lengths per second that the pick-up is moving, one can view sliding the pick-up point as a ring modulation splitting each harmonic, f_k, into two partials, $f_k \pm 0.5\ k\ s$. Figure 4 shows a sonogram of a pick-up making discontinuous speed increases. Notice that the higher harmonics spread more than the lower in this linear frequency plot. This phenomenon is short-lived, however, if we are bounded by the string end points. It is convenient to contrive a theoretical placement of the pickup point beyond the ends of the string by extrapolation of the standing wave on the string. To generalize pick-up point motion we must define what it means to place the pick-up somewhere off the end of the actual string, that is, to extend the string in some way without discontinuity of the traveling waves in the delay lines. We do this by reading the separate upper and lower traveling waves back through the end point filters (here perfect reflections) and around backwards on the other delay line. To read at a pick-up located theoretically at some $1+\delta$, where the string length is 1, and δ is a positive number less than 1, we have the upper displacement, $y^{+}(1+\delta) = -y^{-}(1-\delta)$, and the lower displacement, $y^{-}(1+\delta) = -y^{+}(1-\delta)$. Therefore the composite displacement of the string, y, at pick-up point, $1+\delta$, is as follows:

$$\begin{aligned} y(1+\delta) &= \ \ y^{+}(1+\delta) + y^{-}(1+\delta) \\ &= -y^{-}(1-\delta) - y^{+}(1-\delta) \\ &= -y(1-\delta) \end{aligned}$$

In other words, to obtain a hypothetical displacement beyond the end of the string, just flip the sign and read backwards along the string. This is consistent with the “image method” for computing the displacement from traveling waves (Morse 1936).

Now that we can maintain a continuous motion of the pick-up in one direction indefinitely, we may consider an arbitrary constant *speed* and *acceleration* of the pick-up point along the string. Figure 5 shows a string loaded with two harmonics at

1 kHz and 5 kHz with an accelerating pick-up point to produce expanding glissandi. In this four-second example, the pick-up point speed accelerates from 0 to 1000 string lengths per second. Since a string length is half the wavelength of the string fundamental, the downward gliding frequencies make it half way to DC. (There are also some aliasing sidebands in the picture relating to the error of linear interpolation.) In the four seconds of sound depicted in Figure 6, the pick-up point is being modulated sinusoidally in placement at a rate of 5 Hz, but its deviation from center is 10 string lengths. Where the double vibratos are at their furthest separation is when the pick-up point is flying past its center point of modulation at its greatest speed. Notice that the double vibratos in the upper harmonics cross over each other.

Figure 5.

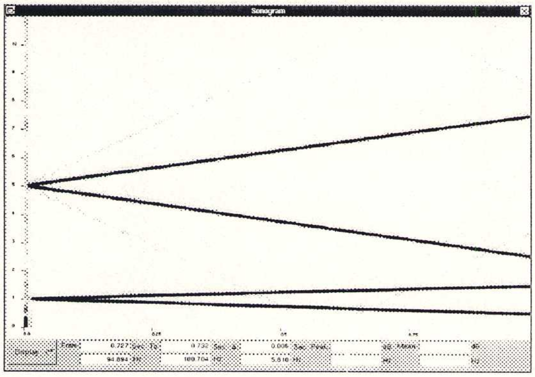
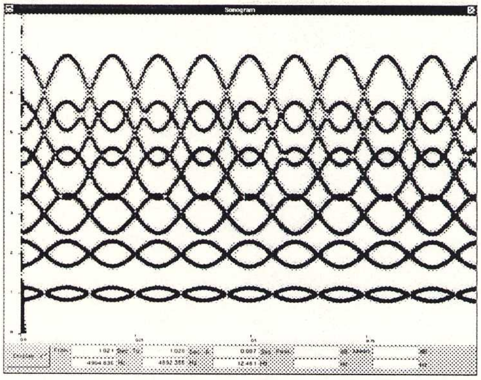


Figure 6.



3. Modulating the Pick-Up Position at Audio Rates

The next logical step is to modulate the pick-up point at audio rates. We define the pick-up point modulation as follows: $x_p(t) = x_{p0} + d \sin \omega_{xp}\, t$, where $x_p(t)$ is the resultant pick-up point placement measured in string lengths; x_{p0} is the center point of modulation; d is the deviation measured in string lengths; and ω_{xp} is the frequency of the pick-point modulation in radians. A closed form expression for the resultant spectrum heard from this pick-up can be found as follows:

$$\begin{aligned} y(t) &= \sum_{k \varepsilon H} y_k(t), \text{ where} \\ y_k(t) &= A_k J_o(I_k) \sin \phi_k \cos \omega_k t \\ &+ \sum_{\substack{i=1 \\ i\,\text{odd}}}^{\infty} A_k J_i(I_k) \cos \phi_k \left[\sin(\omega_k + i\omega_{xp})t + \sin(\omega_k - i\omega_{xp})t \right] \\ &+ \sum_{\substack{i=1 \\ i\,\text{even}}}^{\infty} A_k J_i(I_k) \sin \phi_k \left[\cos(\omega_k + i\omega_{xp})t + \cos(\omega_k - i\omega_{xp})t \right] \end{aligned}$$

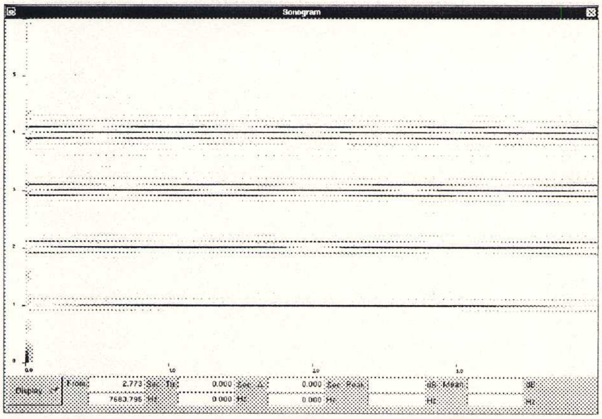
where k is the harmonic number; H is the set of harmonics on the string; y_k is the contribution to the total spectrum resulting from the k^{th} harmonic; A_k is the amplitude of the k^{th} harmonic; $I_k = \pi k d$ is the modulation index; $\phi_k = \pi k x_{p0}$ is the phase offset determined by the position of the modulation center point; ω_k is the frequency of the k^{th} harmonic in radians; and ω_{xp} is the radian frequency of the pick-up point placement modulation.

This expression says that around each of the harmonics on the string, ω_k, sidebands are produced at plus and minus multiples of the pick-up modulation frequency, ω_{xp}, and that their magnitudes are dependent on Bessel functions of the first kind operating on an index based on d, the deviation of pick-up point modulation. These equations differ from ordinary FM, however, in that the odd sidebands are scaled by $\cos \phi_k$ and the even sidebands are scaled by $\sin \phi_k$, where $\phi_k = \pi k x_{p0}$ is, in effect, the relative position of the pick-up modulation center point measured along the standing wave components of each of the harmonics on the string. When x_{p0} is at a harmonic node, that is at a position n/k on the string, where n is an integer and k is the harmonic number, then $\sin \phi_k$ evaluates to zero and the even harmonics are zeroed out. On the other hand, when x_{p0} is at an anti-node of a harmonic, that is, at a position $(n + .5)/k$ along the string, $\cos \phi_k$ evaluates to zero and the odd harmonics are zeroed out. Sliding the center point between a node and anti-node produces a gradual exchange of energy from the odd sidebands to the even sidebands.

Figure 7 illustrates a four second sound created by sliding the center point of pick-up modulation from one end of the string to the other over the duration of the tone. The fundamental frequency of the string is 1000 Hz. Four harmonics were loaded onto the string at start-up. The modulation frequency was 100 Hz and the deviation was 0.1 string lengths. There are more side bands around the upper harmonics due to the greater effect of the deviation distance on the shorter wavelengths. Observe how the odd and even sidebands fade in and out as the center point of pick-up modulation glides over the nodes and anti-nodes of the various harmonics.

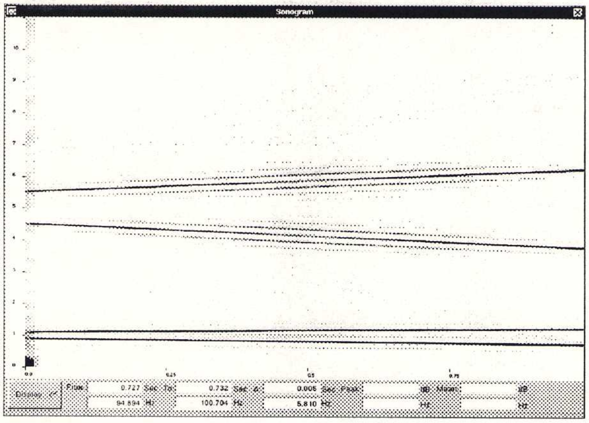
Figure 8 shows how speed and modulation can be combined. Here the first and fifth harmonics were loaded onto a 1000 Hz string at start-up time. The center point of modulation accelerated across the string from a speed of 200 string lengths per second up to 500 string lengths per second. Meanwhile, with a modulation frequency of 200 Hz, the deviation ramped from 0 to .1 string lengths back down to 0 over the course of the sound. Note that we lose the nice feature of individual control of odd and even sidebands when using speed or acceleration in combination with modulation.

Figure 7.



From: 2.773 Sec To: 0.000 Sec Δ: 0.000 Sec Peak: dB Mean: dB
7683.795 Hz 0.000 Hz 0.000 Hz Hz Hz

Figure 8.



From: 0.727 Sec To: 0.732 Sec Δ: 0.005 Sec Peak: dB Mean: dB
94.894 Hz 100.704 Hz 5.810 Hz Hz Hz

Conclusion

While it is true that the results in the simple case of the ideal string with perfect reflections at the end points could be duplicated with traditional FM means, this research brings up new possibilities of combining FM and physical modeling techniques. There are many possibilities when a low pass or other system filter is placed at the end of the string instead of the perfect reflection. Many other model parameters can be similarly modulated, unlike the corresponding parameters of natural instruments, such as string length, bridge coupling, body resonances, and so on. Waveguide models can be enriched by a modulating pick-up point, while on the other hand, traditional FM instruments can be rejuvenated by the loop filtering methods of physical modeling and waveguide synthesis.

References

Momose, H. *Sonogram: An Acoustic Signal Analyzer/Editor.* Software for NeXT, version 0.90(Beta), 1991.

Morse, P. M. *Vibration and Sound*. Published by the American Institute of Physics for the Acoustical Society of America, 1976. (First edition 1936, second edition 1948).

Smith, J. O. and P. R. Cook. "The Second-Order Digital Waveguide Oscillator," elsewhere in this proceedings. See in particular the references at the end of that article regarding the digital waveguide approach.

Smith, J. O. "Waveguide Synthesis Tutorial," to appear in the *Computer Music Journal*, MIT Press, Spring 1993 (est.).

Low Piano Tones: Modeling Nearly Harmonic Spectra with Regions of FM

Scott A. Van Duyne

Center for Computer Research in Music and Acoustics (CCRMA)
Music Dept., Stanford University, Stanford CA 94305
email: savd@ccrma.stanford.edu

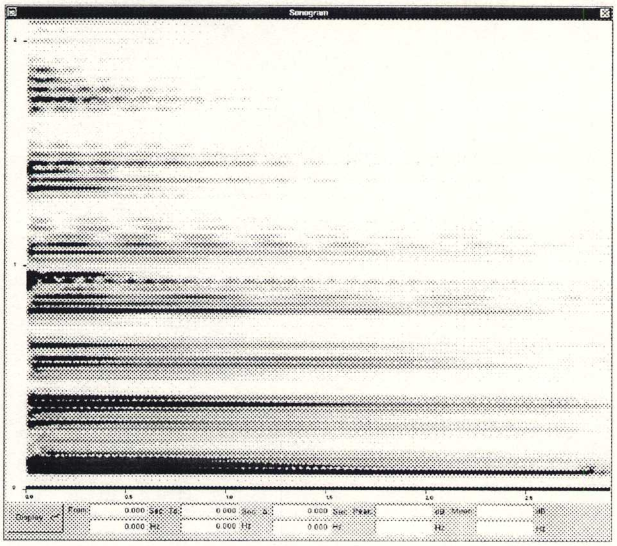
Abstract

The stretching observed in the harmonics of the low piano tone is critical to its timbre. By taking advantage of the natural nulls in the low piano spectrum every eight or nine harmonics, a good FM synthesis of these sounds can be produced with an small set of FM oscillator pairs with carrier and modulator frequencies tuned to create a dovetailed piecewise stretching of the partials.

1. Analysis/Synthesis Model of the Spectral Structure

There are two fundamental elements to the spectral structure of low piano tones which work together to create their characteristic sound: First, the magnitude spectrum is arranged in groupings of eight or nine partials. These groupings are not related to any kind of formant phenomenon, but are apparently the result of the hammer strike position on the string. The hammer position on the low piano strings has traditionally been placed very carefully at some point near 1/8 or 1/9 the distance across the length of the string. The acoustics of such a system dictates these kind of near nulls in the string spectrum at harmonics which are near multiples of eight or nine. The sonogram in Figure 1 shows these groupings in a recorded low piano tone. The horizontal axis is time. The vertical axis is frequency (0 Hz to 2000 Hz). Lightness and darkness indicate the relative magnitudes.

Figure 1. Spectral Evolution of A0

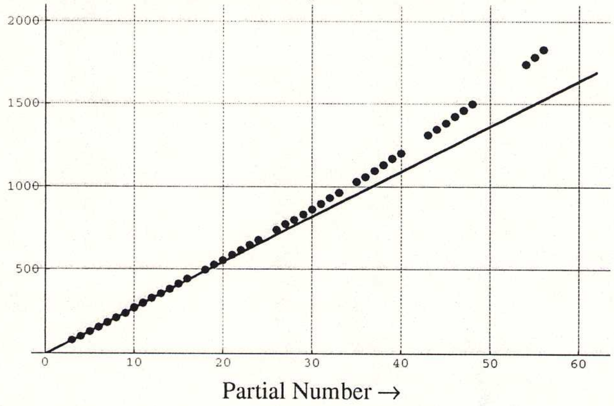


From: 0.000 Sec To: 0.000 Sec Δ: 0.000 Sec
0.000 Hz 0.000 Hz 0.000 Hz

Second, the harmonics of the low piano string are in fact not harmonic, but instead stretched at a fairly linear rate throughout the spectrum. The stiffness of thick piano strings in the lowest register causes the sinusoidal components of the traveling wave on the string to travel faster when at higher frequencies, thereby spawning harmonic resonances in the string at ever increasing partial spacings through the spectrum. Stretching was measured for A0 (the lowest note on a conventional piano) at a rate of 10 Hz over 1000 Hz of spectrum. For example, near the A0 fundamental, the separations between partials are near 27.5 Hz. At a height of 1000 Hz in the spectrum, the separations are near 37.5 Hz, and so on. For the note A1, one octave above A0, the rate of stretching was considerably lower: about 3 Hz per 1000 Hz. These rates, of course, vary from piano to piano. Figure 2 shows the stretching for A0.

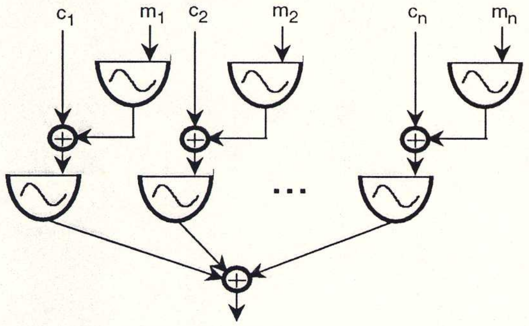
Figure 2.

Measured Partial Frequencies in Hz for A0



The synthesis of low piano tones has been a complex problem due to stretching harmonics. However we can take advantage of the spectral groupings in the piano spectrum to build an FM model of piecewise linear partial separations. To model a string with fundamental f_0, assign one FM carrier/modulator pair to each of the groupings. Let the carrier and modulator frequencies for the i^{th} FM pair be given as c_i and m_i. Choose m_1 to be f_0 and c_1 to be 5 f_0. With the proper choice of FM Index, this pair can be used to model a grouping of eight or nine partials. To obtain a piecewise stretching of the partials, select the parameters of the second FM pair to have a modulator slightly larger than m_1, and a carrier c_2 placed eight partials above c_1. Continuing this way with as many pairs as desired, we can construct a piecewise stretched spectrum arranged in groupings. We just require $m_1 < m_2 < m_3 < ... < m_n$ to model the stretching rate. Care must be taken to dovetail the pairs together with a proper choice of the carrier frequencies. We need $c_i = c_{i-1} + 4\ m_{i-1} + 4\ m_i$ to insure that the fourth sidebands of each pair match frequency with the adjacent pair. Note that there is no longer any meaningful c:m ratio for pairs above the first. The Index should be chosen not to large, to avoid de-tuned overlap of sidebands from adjacent pairs, and, on the other hand, not to small, in order to produce enough sideband presence to maintain a coherence of the spectrum. Mapping the amplitude envelope to the Index envelope seems a good first approximation here. Figure 3 shows an outline schematic of the instrument.

Figure 3.



The experimental results show that a very good low piano tone can be made using only four or five of these additive FM pairs. Even with no stretching, i.e. with $f_0 = m_1 = m_2 = ... = m_n$, a reasonable approximation of a piano is possible. However, with just the slightest piecewise stretching introduced, a naturalness and life is infused into the tone color which is hard to come by in an exactly harmonic spectrum.

2. Construction of Decay Envelopes

The choice of amplitude envelopes for the FM pairs is critical in the production of a believable piano sound. The decay envelope of the piano tone is generally exponential. However, the rate of decay is faster toward the beginning of the tone than later on. This effect can be modeled by making a linear combination of two exponential decays as follows:

$$\mathrm{Amp}(t) = \alpha_1\, e^{(-t/\tau_1)} + \alpha_2\, e^{(-t/\tau_2)}\, ,$$

where t is time, τ_1 and τ_2 are the time constants of decay, and α_1 and α_2 are the relative scalings of each envelope component. We use a relatively short τ_1 to model the decay rate in the first one or two seconds after the attack, and a longer τ_2 to model the "steady-state" decay rate.

There are several other considerations in the construction of appropriate envelopes: (1) Higher frequency components of the piano tone die out more quickly that lower frequency components, necessitating separate parameters for each of the FM pairs used. Also, (2) louder attacked notes begin with a brighter spectrum, requiring more energy in the higher pairs at the attack. In addition to the decay envelope differences, (3) the cut-off release time is much longer for A0 than for A1. The slow release time of the lowest piano notes is an important part of their timbral signature. (4) The attack portion of the amplitude envelope of each FM pair is constructed using a rise-rate method where the actual time before the highest amplitude is reached depends on how high the envelope must rise. This naturally rounds out the attack a little by de-synchronizing the attack time over the spectrum. In addition, (5) a constant "speaking-threshold" segment is introduced before the rise time portion to bring the amplitude up from 0 to just audible. This segment, if the parameters are chosen right, produces a characteristic "thump" and broadens the attack.

3. Fine points: Beating in the Partials

Some periodic amplitude fluctuation in the middle and higher partials on the order of 2 to 4 Hz is visible in Figure 1. This was modeled by adding a sinusoidal amplitude modulation component to the Index envelopes of the FM pairs. Modulating the Indices instead of the amplitudes of the pairs attempted to give a variety of fluctuation levels for the different partials in each pair. The frequency of this modulation was given at higher rates for pairs which contributed to higher spectral areas, as was the general trend in the analyzed spectral data. While the resultant spectral pictures of these attempts were more like the pictures of real data, the auditory differences were negligible and otherwise only added complexity to a simple, elegant synthesis algorithm.

Conclusion: What now?

It is now evident that excellent models of nearly harmonic spectra can be built by separating the spectrum into piecewise regions of equally spaced partials, each region modeled with a carefully tuned FM pair. Using the same procedures, an entire class of synthesis algorithms become available for musical timbres with slowly changing rates of partial stretching or partial contracting over the spectrum.

References

Chowning, J. "The Synthesis of Complex Audio Spectra by means of Frequency Modulation." *JAES* 21(7), 1973.

Schottstaedt, W. "An Introduction to FM." Unpub. 1992.

Holm, F. "Understanding FM Implementations: A Call for Common Standards." *CMJ* 16(1), 1992.