UNIVERSITY OF NORTHERN COLORADO
MUSIC TECHNOLOGY

AN OVERVIEW OF

PHYSICAL MODELING SYNTHESIS

Its Origins and Applications

by

Doug Tapia
University of Northern Colorado
Music Technology

What Is Physical Modeling?

Physical Modeling is an exciting and fairly new method of electronic sound synthesis that offers great potential to both the musician searching for the most convincing real-world sound emulations, and to the musician searching for unique, never-before-heard sounds. Technically speaking, Physical Modeling is a complex computer algorithm that is used to describe the sonic and performance behaviors of an actual acoustic instrument.[1]

Modeling technology in one form or another has been around since the early 1970s — about the time that engineers first started playing around with digital filters, a major ingredient of modeling. The technology started to come of age in the early 1980s, with the most familiar work — to musicians, anyway — being the Karplus/Strong Plucked String Model.[2] This model, along with research by scientists at Stanford's Center for Computer Research in Music and Acoustics (CCRMA) Labs and the University of California at Berkeley's Center for New Music and Audio Technology (CNMAT) have helped to usher in one of the most exciting and promising new technologies in the history of synthesis.

Why All The Excitement?

Musicians and scientists alike are excited about physical modeling because it is unlike any musical synthesis technology that has ever existed. Physically modeled instruments offer the musician a much greater degree of control than is available through more traditional sound generation schemes.

The physically modeled instrument can respond to performance gestures just as an acoustic instrument would. For example, when a wind player lets up on his or her breath pressure, the note being played tends to go a little flat. Likewise, when the physical model of a wind instrument is played with less "breath support," it too will go a bit flat.

More traditional synthetic instruments can be made to mimic this behavior, however subtle performance gestures such as tonal shadings, scoops, and changes in vibrato rate and depth are difficult to accomplish in real time in any sort of convincing fashion. The reason that physically modeled instruments are so much more adept at these sort of subtle (and not so subtle) performance gestures is due to the very nature of how a physically modeled instrument creates sound.

---

[1] Marans, p. 100.

[2] Marans, p. 100.

So How Is Physical Modeling Different?
(a brief digression into the history of synthesis)

To understand how physical modeling is different from other, more "traditional" forms of synthesis, we must first examine how these other forms of synthesis.

Analog Synthesis. This is the technology upon which the synthesizer industry was born. Analog Synthesis operates, at it's most basic level, by taking a handful of waveforms, such as sinewave, sawtooth, and squarewave, and manipulating them with other waves, filters and envelopes. Analog synthesis excelled at the expansion of the sonic palette, through the creation of new timbres and never-before-heard classes of sounds. Classic sounds, such as filter sweeps, oscillator sync, and untold numbers of sci-fi bleeps, blurps, and tweebles remain as lasting icons, firmly embedded in our popular culture. Analog synths, however, failed miserably when it came to the emulation of acoustic instruments.[3]

Frequency Modulation (FM). Yamaha introduced the first FM synthesizer, the DX7, as "the next big thing" back in 1983. The DX7 single-handedly redefined the modern synthesizer: the myriad of buttons and knobs commonly found on analog synths was replaced by a single data slider; an LCD display, the ADSR envelope generator, became the far more flexible rate and level envelope; and the very core of sound generation — analog square, sawtooth, and triangle waveforms — was replaced by arrangements (called algorithms) of carriers and modulators that could produce hundreds of waveforms of varying, complex spectra. It had a clean, edgy timbre that was very different from analog synths.[4]

The DX7 could not only generate new classes of sound, but could also emulate some acoustic instruments more accurately than analog machines could. Still, it wasn't too good at that particular task, and its clean, digital sound eventually became tagged as lacking the analog "warmth" that musicians had come to know and love.

It is interesting to note that in many ways, the synthesizer industry is still rebelling against the Yamaha DX7 and all its conventions. There has recently been a "vintage synth" craze that has catered to musicians seeking the "analog warmth" of a bygone era. Also, many have found the tiny LCD-single data slider to be a less than satisfying way to program an instrument, and have found that this same

[3]Marans, pp. 100–101.

[4]Marans, pp. 100-101.

combination affords very little real-time control over the sound. The result has been that a few manufactures have added "interactive control surfaces" to their instruments in an attempt to make the playing experience more "analog-like."

Sampling. On the heels of FM came an exciting new technology: sampling. Sampling finally addressed the musician's desire to realistically replicate real-world acoustic instruments. At first it was an expensive technology, limited to the very wealthy, established musicians who could afford to invest hundreds of thousands of dollars in instruments like the Fairlight. Later, as the price of computer memory started to fall, so did the price of samplers, and machines such as E-mu's Emulator began to pop up everywhere.

Sampling was seemingly the ultimate in synthesis, however musicians soon learned otherwise. Samples are static snapshots of a sound. So when a sampled sound is played, the sound is fixed at a particular timbre. In essence, to change a sample's pitch, the sampler is simply playing the digital "snapshot" slower or faster, which plays havoc with the sounds attack, vibrato, and transients.

Hybrid Synthesizers. Nearly every synth on the market these days incorporates multiple synthesis methods, the goal being to overcome limitations inherent in any single sound generation technology. The combination of technologies is found in sample-playback synths, which contain ROM samples that can be processed using a variety of analog synth-style functions, including filters, amplitude envelopes, LFOs, and so on.[5] Other instruments, notably those from Yamaha, combine sample playback with FM. Still others, such as Kurzweil's K2000 series, combine all of these methods, and some that I've not even touched on to provide the most flexibility in sound design possible.

So, Really, What Makes Physically Modeling Different?

Physical modeling is different for a number of reasons. First of all, the above mentioned methods of synthesis have been hardware based. Pure physically modeled instruments are entirely based in software. How is this possible?

In pure science there is an equation for representing any physical action occurring in nature. There is a formula, called the *wave equation,* in which the vibration of an "ideal" string is modeled — that is to say, accurately represented — using mathematically described components, such as the string's tension, linear mass, density, and displacement. The

[5] Marans, pp. 100-102.

equation itself is fairly complex, and its solution is quite involved and computationally intensive.[6] This equation however, can be used to represent any sound producing body, so if the object to be modeled changes, all that is required is to plug in another set of numbers.

This is revolutionary in the synthesizer world, because musicians have always been stuck with the features (if not the sound set) that their instrument originally came with. Perhaps a comparison is in order:

Physical modeling synthesizers are much like the common PC. Most of us buy computers to do a little word processing, a little accounting, and to check our e-mail. The computer handles all of these tasks with equal grace, and if one day we have aspirations of being a graphics artist, we can buy a graphics program and edit images to our heart's content. This is exactly the kind of flexibility that physical modeling offers us: we can do things we never thought we'd have a need to do when we first bought our synthesizer. By contrast, "normal" synthesizers are much like ten keys and typewriters: They are very good at doing the one specific task for which they were designed, but don't function too well outside of their realm of expertise.

Physical modeling is also different in the degree of control over the performance that it offers to the musician. "Physical Modeling provides what is arguably the first truly fully interactive synthesizer instrument — one that can actually be played rather than merely triggered."[7]

This interactivity is a great triumph to musicians seeking more expressive control over the sound they play, but such control does come at a price. Physically modeled instruments take time and practice to learn, and indeed, to master.

A modeled saxophone will sound infinitely better than a sampled sax because sampling does not take into account the fact that a saxophone's sound is more involved than turning on and off a saxophone timbre. In truth, what defines a sax's sound more than the timbre is the manipulation of tone which is idiomatic to that instrument. Timbre is different from one octave to another, attacks are varied, notes are bent into and out of, and there are as many sax tones as there are players and embouchures.

Physically modeling has the ability to reproduce, in real time, all of these variables, but like a real saxophone, one cannot expect to pick up a modeled saxophone and be able to make

[6] Rule, p. 73.

[7] Marans, p. 102.

beautiful music right out of the box. This is because, a player will have to learn to manipulate controls for embouchure, breath support, articulation, pitch, and so on, just like a real saxophone. Without practice, the player of a modeled saxophone has the potential to sound just like an elementary saxophonist on his or her first day of band.

And Why Hasn't This Been Done Before?

As mentioned earlier, modeling involves large equations and an exhaustive amount of computational power. Only recently have the resources been available at a reasonable enough price to make physical modeling worth pursuing at anything more than the university acoustics laboratory level. The last few years have seen processor speeds skyrocket, and the price of computer memory is less expensive than it has ever been. Frankly, there has never been a better time for the exploration of this technology.

Secondly, Yamaha set the stage for physical modeling's debut when they purchased the rights to FM from CCRMA in the late 1970s. Following the DX7's phenomenal success, synth manufactures have been scouring university labs looking for "the next next big thing," and have thus poured millions of dollars into research.

Thirdly, there has been the issue of practicality. Early experiments into physical modeling, though scientifically interesting and full of promise, just weren't usable in a musical context.

Computational speed is of the essence with modeled instruments — a violin that produced a note a few seconds after it was bowed would, of course, be unplayable. One of the main reasons that modeling is finally coming out of the research labs and into the hands of musicians is that processor chips are dramatically faster now than just a few years ago, making it more viable to produce a violin model that will, in fact, respond in real time.[8]

So How Does This All Work?

To create a computer model of a musical instrument, one essentially reduces its physical parts — a violin's bow, body, bridge, and strings, for example — into mathematical formulas; the playing action, such as a bow being drawn across a string, is also mathematically represented. Once the equations are derived, one must plug them into an algorithm (which describes the interaction of these elements) and set the processor to work crunching the numbers. If the instrument descriptions are accurate — that is, if the

[8] Marans, p. 104.

equations contain the right numbers in the right relationships — the computer will be able to construct a model of the "real" instrument. When that model is stimulated — in other words, when the modeled playing action is initiated by a MIDI controller or some such input device — the algorithm will start processing the numbers and the computer will output the appropriate sound. Needless to say, the more accurate (and correspondingly more complex) the model, the more accurate and detailed the resultant sound.[9]

Traditional modeling techniques incorporated direct implementation of the wave equation, which required significant numbers of mathematical operations — and thus, computation time — to execute. This hurdle was overcome in part through the work of Julius O. Smith at Stanford University's CCRMA. Smith's contribution, in which the computations needed to describe the functions detailed in the wave equation are significantly decreased in number, is called the *Digital Waveguide Filter.*

Simply stated, a waveguide (as it is commonly known) describes two waves: a positive one traveling to the right, and a negative one traveling to the left. The method for generating the waves is simplicity itself: a digital delay line. (Since the waveguide actually describes two waves, a bidirectional delay is used.) The instantaneous state of a vibrating string or of the waves in a tube (such as a clarinet bore) can be computed by performing certain mathematical operations on the left- and right-going waves. Waveguides, in effect, afford a great deal of computational economy and simplifies the process of creating the models.[10]

How Is A Physical Model Actually Constructed?

There's a whole lot more to a violin than its strings, and a whole lot more to a clarinet than its center air passage. In order to create an accurate instrument model, as many of the instrument's subtleties as possible must be modeled, as well as the minutiae of the playing action. The modeling process can be broken into about six steps:

Step 1. A block schematic diagram of the instrument that is being modeled is drawn. From this, an algorithm designer will create an accurate modeled representation of each separate part in diagram.

Step 2. The significant features and characteristics of the instrument are analyzed. In the case of the clarinet, this would include measuring the diameter of the bore, the flare of the bell, the spacing and location of the tone holes, the placement and size of the reed, and so on.

---

[9] Marans, p. 106 & 108.

[10] Marans, pp. 103-104.

Step 3. A detailed list of instrument spectra and frequency response is compiled. At this point textbooks on general physics can be helpful. Computer analysis of recorded sounds is also commonly performed. There are programs that will convert the results of the analysis into filter coefficients that can be plugged into the algorithm.

Step 4. The general principles of how the instrument makes its sound is studied next. For example: How much air pressure is required to vibrate a clarinet reed? How does the flare of the bell affect timbre? Does all of the sound come form the bell, or does some sound come directly from the reed as well? Is there noise in the sound? If so, does it come from the bell, or from the tone holes, or from the player's mouth, or all of the above?

Step 5. The playing action is analyzed: This would include a study of the player's embouchure, what happens when the reed is bitten, what role the tongue and lips play, and so forth. Other factors would include the action of the register key, how the tone holes affect the timbre and pitch, what happens when the player blows harder and softer, and so on. Since the ultimate goal is to produce a model that's every bit as expressive as the acoustic instrument, it's very important to detail even the most subtle performance nuances.[11]

So Now What?

Once all this information has been collected and sorted, a development tool such as the Synthkit Algorithm Design Tool is used to construct the actual model. The Synthkit Algorithm Design Tool was developed by Korg Research and Development in Milpitias, California. The Synthkit runs on a Mac OS computer and provides a graphic programming environment for designing and testing synthesis algorithms. The program greatly simplifies the process of creating a physical model.[12]

For the utmost realism, each separate part of our clarinet is represented by its own waveguide, each of which interacts with the others in realistic ways. For the body of the clarinet, only one waveguide is required, as this is mathematically a simple shape. However, the conical bell at the end of the clarinet is vastly more complex and would require many waveguides to accurately represent it, and would thus eat up a lot of processor horsepower.

Algorithm designers have found a more cost way to accomplish this same function. Since

[11] Adapted from Mike D'Amore at Yamaha and Marco Alpert at E-mu Systems.

[12] Marans, p. 103.

the job of the bell is basically that of a lowpass filter, a lowpass filter is employed, greatly simplifying the model by eliminating the need for four or six additional waveguides.

This Sounds Complex. Do I Need A Ph.D. To Understand Modeling?

In reality, physical modeling is quite complex, but any musician who is willing to put in a little time learning the concepts behind modeling, and then practicing playing a modeled instrument can reap great benefits from this powerful new form of synthesis.

Virtually all modeled instruments can be broken down into two main functional sections. The excitation part of the model is what is used to stimulate the model to begin producing sound. On a clarinet model this would be the reed. Due to the nature of wind instruments, which must respond to the player's uneven breath pressure and changes in embouchure, the excitation parts of these wind models are always developed using non-linear mathematical functions. This allows the algorithm designer to mimic and respond to the continuous changes that are produced by human players.

The second main section of the model is the resonator, which is used to model the instrument's body. In our clarinet, that's the center bore, which would be modeled using a bi-directional delay line — the waveguide. Designing the proper resonator is crucial in a model because it will, in large part, determine the instrument's overall timbre. This is where the spectral analysis part of the research is invaluable. It allows the algorithm designer to get a detailed picture of the instrument's format structure. Formats are the major resonant peaks in a given instrument's frequency response; different body shapes produce different sets of formats.

An illustration will help to clarify this: Start by closing your mouth and humming a steady pitch. Now open your mouth, and without changing the way you're humming, form a series of vowel sounds. Notice how each sound is formed as a direct result of the shape of your mouth. Each time your mouth changes shape, the formant series changes, and consequently, so does the vowel sound produced. If we were making a physical model of this, the excitation part would be based on the action of the larynx and the resonator would be the oral cavity.

If you have spent any time with a sampler, you are familiar with the "munchkinization effect," where sounds speed up and slow down as they are pitch-shifted, resulting in unnatural attacks and undesirable changes in timbre. The munchkinization effect occurs because when a sample is played back at a pitch other than its original pitch, the format series in the sound is pitch-shifted as well. So what you end up with is an upwardly pitch-shifted baritone sounding like a chipmunk, and a downwardly shifted soprano sounding like an elephant. In a physical model, the instrument body, and hence its resonant characteristics, does not change as the sound is pitch-shifted; the format series remains the same throughout the entire note range. So when you play a high note on a violin model, the only thing that changes is the pitch, not the body size, the length of the bow or the speed of the vibrato, just like a real violin.[13]

Beyond more realistic sound, the greatest asset of physical modeling is the ease with which performance gestures can be executed. In fact, a large portion of a model is dedicated to interpreting performance gestures — blowing normally versus overblowing, whether an octave shift is due to a register key or a change in embouchure, legato versus staccato tonguing, and so on.

The best physically modeled instruments will give the player enough control to be expressive, but not so much as to be counter productive. A good model will give the player intuitive control over the idiomatic characteristics of the instrument being emulated, without weighting the player down with unnecessary, or uncharacteristic controls. Ideally, a modeled instrument will allow the player a great deal of expression in live performance far beyond what is currently possible using more traditional synthesis.

Physical modeling is not, however perfectly suited to every application, and it is the musician who is able to make the decision when this technology is appropriate and when it is not who will benefit most from these advances.

A Comparison of Sampling and Physical Modeling[14]

Requirement	*Sampling*	*Physical Modeling*
Overall emulation of acoustic instruments	Provides a perfectly detailed but static snapshot	Very good, though fine detail is algorithm-dependent
Solo acoustic instruments	Fair; instrument-dependent	Excellent
Instrument ensembles	Excellent	Not particularly well suited
Drum and percussion sounds	Excellent	Not particularly well suited
Real-world sound effects	Excellent	Not Suitable

[13]Marans, p. 108.

[14]Adapted from Michael Marans.

Comparison of Sampling and Physical Modeling, continued

Requirement	*Sampling*	*Physical Modeling*
Non-real-world sound effects	Very good	Very Good
Real-time performance control	Fair; limited to manipulation of basic parameters	Excellent; allows interpretation of subtle performance gestures
Processing power requirements	Significant, but relatively inexpensive	Intensive, therefore costly
Memory Requirements	Significant, therefore expensive	Minimal
Cross-platform compatibility	Standardized data formats allow data to be interchanged among various samplers; many samplers read sound library disks from multiple manufacturers	Likely to stay platform-dependent for the foreseeable future

BIBLIOGRAPHY

Marans, Michael. "The Next Big Thing." *Keyboard,* Feb 1994, Vol. 20 No. 2, 98 – 118.

Rule, Greg. "Keyboard Reports: Korg Wavedrum." *Keyboard,* March 1995, Vol. 21 No. 3 72 – 78.

Afterword

This manual was set using a Macintosh computer and WriteNow 4.0. Text is set in Adobe Garamond. The information presented in this guide is not guaranteed by The University of Northern Colorado or Coda Music Technology in any way. The information presented in this guide is subject to change without notice.

The author would like to express his gratitude to Joe Bryan, former system architect and senior engineer for the Korg OASys Project, without whose assistance this paper would not be possible. Mr. Bryan is currently at Chromatic Research, accessible via the web at http://www.chromatic.com and http://www.mpact.com.

The University of Northern Colorado Music Technology Center is located on the UNC campus in beautiful Greeley, Colorado. The center serves as a support infrastructure for the School of Music, which incorporates technology into many aspects of the curriculum. Students, faculty, staff and community members alike make use of the center on a daily basis. For more information on the Music Technology Center, visit our home page at
http://arts.unco.edu/musictech/
or write us at the address below.

Music Technology Center
Fraiser Hall
University of Northern Colorado
Greeley, CO 80639