December, 2003

# Live Performance Interaction for Humans and Machines in the Early Twenty-first Century: One Composer's Aesthetics for Composition and Performance Practice

Brian Belet, *San Jose State University*

# Live Performance Interaction for Humans and Machines in the Early Twenty-First Century: One Composer's Aesthetics for Composition and Performance Practice

Dr. Brian Belet
School of Music and Dance
San Jose State University, San Jose, California, 95192-0095, USA

## ABSTRACT

Meaningful real-time interaction between human performers and computer processing is an important aesthetic issue for many composers. With the advent of computer systems that are actually fast enough to permit real-time algorithmic sound realizations based on the analysis of live performance data the issue now facing composers is how to utilize these tools in a meaningful and artistic way.

*Still Harmless [BASS]ically*, composed in 2000 for electric bass and Kyma, is an experimental environment designed to test real-time performance interactive issues. This composition has been performed numerous times by several bassists, and lessons learned from this project have been used for later work. *Lyra,* composed in 2002 for violin and Kyma, is an environment in which the acoustically generated sounds and the computer generated and processed sounds are fully and mutually dependent upon each other for a unified ensemble performance. All of the computer music layers are generated in real time using the violin music as direct audio and as analyzed input data for processing, resynthesis, and other parameter control. The violin music is also affected by the computer music output through performance instructions that invite response and improvisation. The violinist is also able to control macro time and gestural synchronization with the computer sound layers. The aesthetic goal is to create a performance environment that permits maximum flexibility for the human performer with a great deal of linear independence for both violin and computer, while still maintaining a very high degree of unity and ensemble interaction. This work has been performed numerous times by several violinists, and the aesthetics and practicality of this interactive performance environment are deemed successful.

## 1. INTRODUCTION

Technology influences all art, and therefore all music, including composition, performance, and listening. It always has, and it always will. For example, technical developments in materials, mechanics, and manufacturing were important factors that permitted the piano to supersede the harpsichord as the primary concert Western keyboard instrument by about 1800. And with each new technical development new performance issues have been introduced. Piano performance

technique is quite different from harpsichord technique, and composers responded to these differences with new music ideas and gestures. The multiple relationships between technology and composer and performer are dynamic and of paramount importance to each party. And a true consideration of any aspect of music requires that all three areas be examined. This has always been a part of music, and so these relationships are inherently important within computer music. The difference is that electronic technology has caused a fundamental change for all aspects of music, a difference that is as pivotal in the history of Western music as was the shift from oral to written preservation of music over a thousand years ago, and then also the accessibility provided by printed music five hundred years ago. In computer music all parties are always acutely aware of the presence and influence of machine technology in both the visual and audible realms.

One problem integrating human performers with electro-acoustic music has been the aesthetic dilemma of who serves what (or what serves whom) in performance. The static paradigm of live performance with prerecorded tape can offer a performer bragging rights if s/he is successful in synchronizing an exposed attack at performance time 13.23; yet it also creates a drastically unfair performance environment in which the live performer is the only party who can be at fault when such a simultaneity does not occur. In this situation the live performer is usually set up to serve the demanding inflexibility of the prerecorded music. For many performers this is not an enjoyable performance environment; it clearly does nothing to advance the mutual give and take needs of ensemble music making since the tape music is inflexible. In contrast, several computer music languages and systems now provide genuine interactive performance environments in which more equal balances between human performer and machine can be created. It is one task of composers to adapt their creative processes to explore and further refine this area of music making.

Two recent compositions by this author were designed to explore the issue of meaningful real-time performance interaction between live performer and computer: *Still Harmless [BASS]ically*, for electric bass and Kyma (2000) was composed for Jim McManus, and it has served as an experimental platform to test and develop refinements to the interactive process by several bassist (including the author, who has recorded it as bassist), with over twenty performances to date in concerts and festivals throughout the United States and in Europe. *Lyra*, for violin and Kyma (2002), was composed for Patricia Strange, who has recorded the work and performed it over ten times to date. Her feedback, and that of the other violinists who have performed the work, has provided valuable data for additional refinements and further work. Some of the interrelated performance and composition issues within these two compositions are presented in the main body of this paper, with referenced sound examples.

## 2. AESTHETICS

Art reflects its own time and place. If art needs to have a function within society then this is one (another being that art can forge the way for society's future). Western music history has shifted from oral to written and now to electronic means of creating and preserving information over the past two thousand years. And each paradigm shift has generated new compositional and performance issues, with subsequent changes demanded of the listeners. Electro-acoustic music is a twentieth century development, and its aesthetics are linked with the prevailing and constantly changing views of society towards machines in general. The twentieth century traversed an incredible range of machine use, which created a dualistic social dynamic. Machines were viewed both as time saving miracles with the promise of creating an easier life for each of us, and simultaneously as diabolical threats to our central humanity. These linked and opposing themes of good and evil were portrayed in various ways in numerous films, books, and theatrical productions from the early part of the century (reacting to the social changes created by the Industrial Revolution) through the post-Orwellian fears of the 1990s. And music composition, performance, and listening have all been affected by this dichotomy. Since most music involves technology, whether it is the intricate mechanism of a piano, the choice of woods and carving algorithms for a viola, or the machined holes and keys of a flute, the twentieth century reactions to electronic machines in music were primarily a factor of larger social concerns rather than separate musical issues.

Twenty-first century computer music does not need to address the question of using machines, and specifically computers with real-time processing capabilities, in music, but rather how they should be used. 'One cannot turn one's back on the most significant technological breakthrough in history without risking irrelevance to that history' (Garnett 2001: 32). And interactive relationships between humans and machines are relevant musical concerns just as they are in a larger social context.

For better or for worse computer music aesthetics can be viewed as 'an extension of the Western art practice of giving priority to the abstract organization of its material and forms' (Truax 2000: 126). On an individual listening basis music still has to identify itself through its own sounds, yet this subjective value judgment process is difficult to codify and discuss. Since the early shift to detailed written notation Western music composers have had several centuries to investigate multiple levels of structure that are designed to relate only to themselves as musical materials.

Due to the technical ability to design sounds both on extreme micro and macro levels computer music permits a paradigm in which 'sound and structure become increasingly inseparable' (Truax 2000: 119). Purely instrumental and vocal music separate sound and structure, as timbre is conceptually distinct from pitch and time, both in composition and in performance. Computer music integrates these aspects since elemental sound design is not a separate concept from larger

structures. All levels of music must be composed and eventually represented as binary data, arguably 'the most abstract and context-free form it has ever taken' (Truax 2000: 120).


## 3. STATIC PERFORMANCE SYSTEMS

The performance paradigm of live performer competing with static tape originated as a necessity of technical limitations in early electro-acoustic music. There are physical limitations for all modes of performance, both human and machine. Composers have always been quick to adapt to their given limitations, and then to push the edge of these same limitations through compositional experiments. The energy that exists between composers' desires and the existing physical limitations of performers and instruments is one of the motivating factors for the composition process, and good performers share in this exciting dynamic. As long as the technology prevented computer sounds from being truly interactive with live performers, the situation was considered acceptable to composers and performers alike as a simple recognition of reality. And a new compositional aesthetic developed that favored this performance environment. Whether positive or negative, this paradigm does place the live performer at a distinct practical disadvantage since virtually all performance errors are viewed by the audience as errors by the live performer. The tape or other fixed electronic sound media is relentless and unforgiving as it simply plays on.

Relatively recent developments in both software control and hardware processing have created new performance paradigms. While the change from static to dynamic performance interaction has been underway for more than a decade, it is really just within the past three to five years that significant enough changes have occurred to create new performance environments that genuinely permit an equality between live and machine performers to create new and vibrant modes of ensemble interaction.


## 4. INTERACTIVE PERFORMANCE SYSTEMS

The terminology used in computer music is fluid and therefore often confusing. This is to be expected in an area of music that is so dynamic. The terms real-time music, interactive performance, and even artificial intelligence have been used and misused so often that they have been rendered virtually meaningless as they have morphed into academic jargon. Guy Garnett offers a helpful clarification: he defines interactive computer music as a sub-genre of 'performance-oriented computer music' in which a performer controls the computer music sounds in some way (Garnett 2001: 21). And this larger genre includes all computer music that includes a strong performance component. Performer involvement in the machine process runs the gamut from quite simple activity (e.g., pushing PLAY for prerecorded playback, triggering predetermined processing modules, and multi-channel sound diffusion) to more sophisticated

levels of musical interaction (e.g., mapping performer gestures as input data, and analyzing performance audio for real-time resynthesis and as processing control data). Interactive computer music is still a very large genre, one that has changed throughout the era of computer music, and one which itself includes many sub-definitions and sub-genres.

Earlier limitations of computer technology restricted interaction to relatively simple levels. Compositions using only delay buffers, reverb, and pitch shifting have long been labeled as interactive works. The computer limitations demanded aesthetic compromises by composers and performers, and we learned to just deal with the situation (and to also overlook the loose use of language). In this limited interactive environment the human performer remains the more sophisticated party, with faster musical reactions and usually more interesting musical results than those provided by the machine processing. Early systems labeled real-time portrayed significant audio latency problems when processing and resynthesizing live performance data. All computing requires some finite time to process data, and this latency was a byproduct of the slower processing speeds of CPUs, with associated data I/O and buss bottlenecks. Again, compositional aesthetics and techniques adapted to deal with these physical limitations (soft attacks, heavy reverb, and a Gaussian granulation envelope can cover quite a bit of latency!).

This author's early work with real-time systems emphasized algorithmic real-time programming and synthesis (Belet 1991), followed by a combination of real-time synthesis and live interaction, but with practical emphasis still on programming issues (Belet 1992). Current computer technology now permits interactions that are more technically sophisticated and therefore more musically viable. Fast software coupled with dedicated DSP hardware has reduced the latency problem to near inaudible levels for the most complex analysis and processing algorithms (currently three milliseconds for FFT analysis and resynthesis) and only a single clock cycle for more traditional direct audio processing (e.g., nonlinear wave shaping, granulation clouds, frequency and amplitude modulations, and stochastic algorithms). Depending on the algorithms employed some noticeable latency can still exist due to the windowing overlap process of live audio signal analysis. Perhaps this will never be completely erased, and perhaps this is not inherently negative. All voices and instruments have physical limitations that composers and performers learn to work around and minimize, and perhaps computer latency will remain a compositional challenge in real-time systems. Composer and performer aesthetics are rapidly changing to once again take full advantage of these technological developments. What was accepted for performance latency times a few years ago is no longer acceptable today. Performers have noticed the benefits provided by the new performance environments, especially noticing the equality provided by real-time computer analysis and resynthesis, which enables the computer processing clock to be controlled by the live performer. This capability invites composers to create new models for composing; models which are flexible and which permit more performer control of the overall shape and development of the music. Audiences are also

invited to modify their conservatory preserved concert aesthetics as they encounter a unique event with each performance of a work. This requires dedicated active listening by a knowledgeable audience with an understanding of both deep and surface musical structures, as was necessary when Mozart and Beethoven performed their own piano sonatas, and which is still necessary today when listening actively to jazz and progressive rock. This is the classical tradition, so interactive computer music is actually adhering to a long established and respected cultural paradigm.

## 5. KYMA.5 SYNTHESIS AND PERFORMANCE SYSTEM

There is considerable documentation available outlining the details of the Kyma digital synthesis system, including seminal writings by developer Carla Scaletti (Scaletti 1989, 2000, 2002). Kyma is a computer language, with its own elements and organizational grammar, which sets it apart from a computer utility that is designed to perform a predefined function. Contemporary computer music languages are designed to be open to user experimentation (including uses not originally anticipated by their developers) and can be easily extended through ongoing software enhancements. These languages are of paramount importance to composers and performers due to their flexibility and sophistication. 'Although they may never command the same market share as utilities, they have had a longer-lasting and deeper influence on the evolution of music' (Scaletti 2002: 69).

The object-oriented structure of the Smalltalk-80 programming language is reflected throughout Kyma, from low level sound and algorithm design to macro structure and the performance interface. Everything in Kyma is defined as a Sound, with a nested user-defined structuring of subSounds and superSounds, with which 'the composer creates a universe of sounds and a set of assertions giving the relationships between those sounds' (Scaletti 1989: 62). This relationship is recursive and it permits the composer to work freely in and across the previously disparate areas of theory, composition, and performance realization. 'Within the Kyma environment the composer defines both *compositions* and a *theory* of composition' (Scaletti 1989: 63). Composers are the theorists, using the established scientific sense of the term: they propose musical theories for which specific compositions are generated to test these theories. The author's program COMP1 was written as a computer music theory that formed the basis for compositions from 1985 to 1990, and COMP2 is a more refined theory that has been continually modified since 1992 to generate additional compositions (Belet 1991, 1992). Within Kyma these theories have included sets of definitions of sound classes and algorithms relating these sound classes to each other, to the human performer, and to the realization of the composition in live performance. Kyma.5, released in 2001, includes major changes on all levels of the program that offer a working environment to address these musical issues in a flexible and sophisticated way

for both composers and performers. Some of these features are discussed below in the context of specific compositions.

## 6. COMPOSITIONS

*Still Harmless [BASS]ically* was originally composed as a static work for electric bass and tape (*[BASS]ically Harmless*, 1994). The work has been revised numerous times as the technological capabilities of Kyma and the author's programming skills have increased. The most recent and drastic revision transformed the static work into an interactive performance environment (renamed *Still Harmless [BASS]ically*, 2000). In this manifestation most of the computer music is generated during performance directly from the bass music, which is fed into the Capybara hardware as a direct audio input signal and analyzed continuously within Kyma. Sounds are temporally organized within Kyma.5 using the TimeLine graphical interface, which also serves as a useful performance interface. Sound icons are graphically placed on an arbitrary number of tracks, with each Sound representing a synthesis or processing algorithm (i.e., a separate program running on the Capybara), with its own start and stop times. The macro structure for *Still Harmless [BASS]ically* is displayed on a TimeLine, with a total of twenty-eight time bar graphics (representing Kyma superSounds) scheduled for the composition on Tracks 2 through 8, utilizing both serial and parallel processing times (Figure 1). Track 1 contains five WaitUntil control Sounds, which are described below.
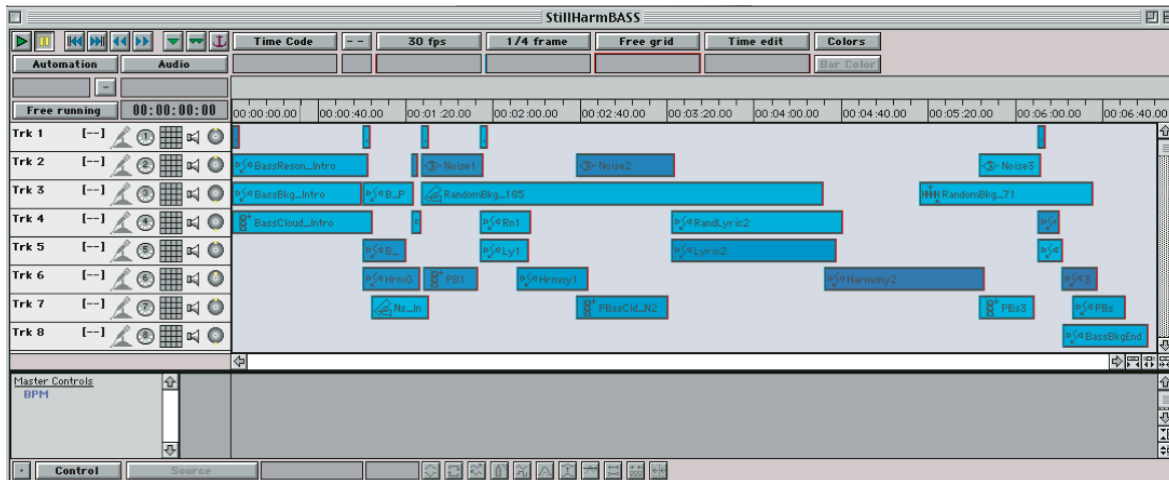


Figure 1. Kyma TimeLine for *Still Harmless [BASS]ically* showing the various computer algorithms organized on various tracks. Interactive Sounds, those utilizing bass audio input for real-time synthesis and processing, include BassResonance, BassBackground, BassCloud, B_Pedal, HarmonyGrain, Noise, Lyric, RandLyric, and Harmony. The two RandomBackground Sounds are separate stochastic structures, and Noise_Intro (Ns_In) is the sole completely determined fixed Sound.

The bassist initiates the overall composition performance as well as several internal sections by starting (and restarting) the computer clock time within Kyma using the WaitUntil Sound, which uses either an amplitude threshold or a specific frequency test to set positive or negative signal controls. The WaitUntil Sounds are placed on Track 1 in the TimeLine for this composition for easy visual identification during performance. The bassist begins with an improvised introduction, and the bass' initial attack creates a positive amplitude control value that starts the TimeLine clock for immediate computer sound generation and synchronization via the WaitUntilAmp Sound (Figure 2).
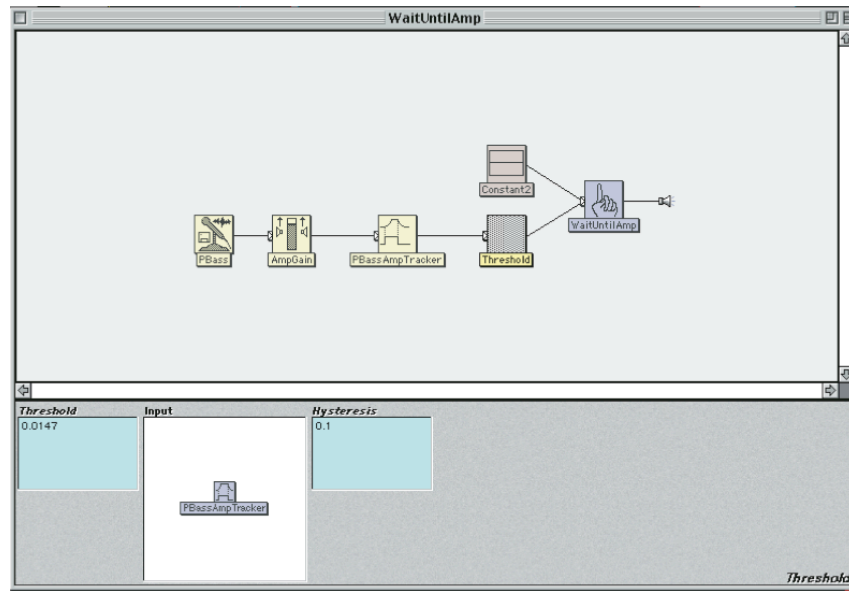


Figure 2. The electric bass amplitude is tracked within a WaitUntilAmp Sound, and a positive control signal is generated when the tracked amplitude exceeds the threshold value.

The bassist also controls the start of several internal sections using the WaitUntilFreq Sound, which tests the bass audio input against stored frequency test values (Figure 3).
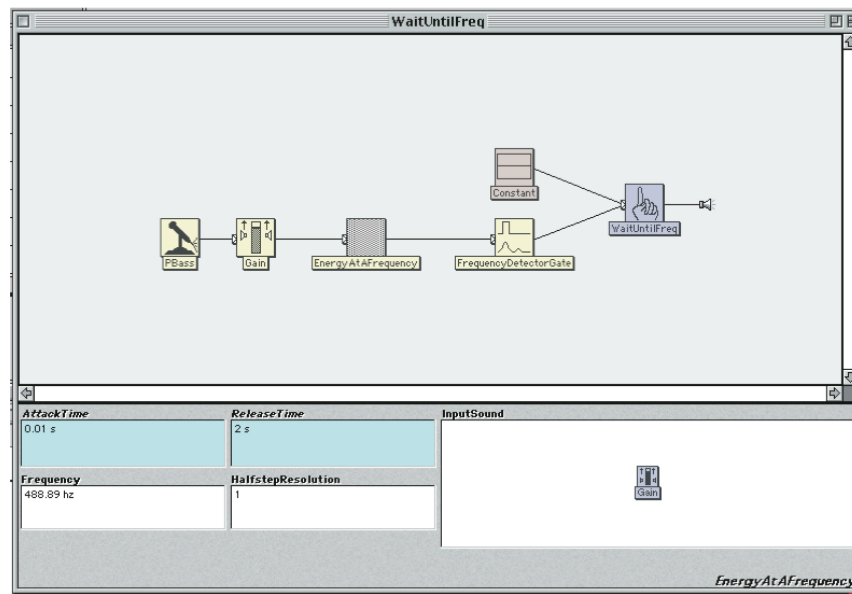
Figure 3. The bass tone B4 (notated in the score as the fifth partial G string harmonic) matches the test frequency of 488.89 hz in this WaitUntilFreq Sound. The positive match generates an output value of 1 from this Sound, which resumes the computer clock time (time resumes at 1.00 within the TimeLine and on the bass score) to synchronize the beginning of the second introduction section.

In *Lyra* (violin and Kyma, 2002) thirteen of the fourteen computer layers are also generated live in performance from the violin music. The single remaining computer layer is a stochastic background commentary that is present for only 3.19 of the 10.30 composition duration. As with the earlier composition, the violinist starts the computer processing, and therefore the performance, using the WaitUntilAmp Sound. This has proven to be an effective and aesthetically pleasing hands-off method of synchronizing the human and machine layers at the beginning of the performance. The composition is unified by having all of the computer music relate to the violin music, both directly and indirectly. There are no pre-recorded or pre-synthesized sounds stored in memory as physical samples or deterministic algorithms. Without the violin playing there is no computer music audio output. The violinist also controls the overall texture of the computer sounds throughout the performance so that density, gestural speed, and energy remain consistent between the two performance sources.

## 7. COMPOSITIONAL ISSUES
Current computer language technology permits composers to construct sounds and sound structures (i.e., computer instruments) that are unique and aesthetically focused for the specific composition being created. Each composition can have its 'own sound-world, a unique physiognomy, which cannot be identified with any other in the world' (Stockhausen 1996: 88).

Stockhausen defines this unique identity for each composition as 'liberated music' (Stockhausen 1996: 100).

The recursive nature of Kyma stems from the object-oriented structure of the Smalltalk-80 programming language, and this permits composers to define meaningful internal relationships between elementary ideas and their resultant structures. These structural relationships are an important concern to many contemporary compositional theorists. 'The sounds are a constituent part of the form' (Stockhausen 1996: 89). This flexible capability invites flexible approaches to composition. Composers can free themselves from the paradigm of fixing strict instructions on paper for a performer to interpret and present in performance. The author's two interactive compositions cited in this paper use composed performance environments with notated gestures that serve as primary time structures and stylistic outlines for the performer to initially follow and then to modify in a variety of ways.

## 8. PROGRAMMING ISSUES

The audio signal from the live performer is routed into the Capybara DSP hardware as a direct audio input. In *Lyra*, this audio signal is processed directly through several algorithms, including granulation clouds, vocoders, multiple random delays, and ring modulation with delayed retrograde output. A RandomLyric Sound, displayed in the Sound Editor mode to display its signal flow, utilizes the violin amplitude data as a processing parameter within a Vocoder synthesis algorithm (Figure 4).
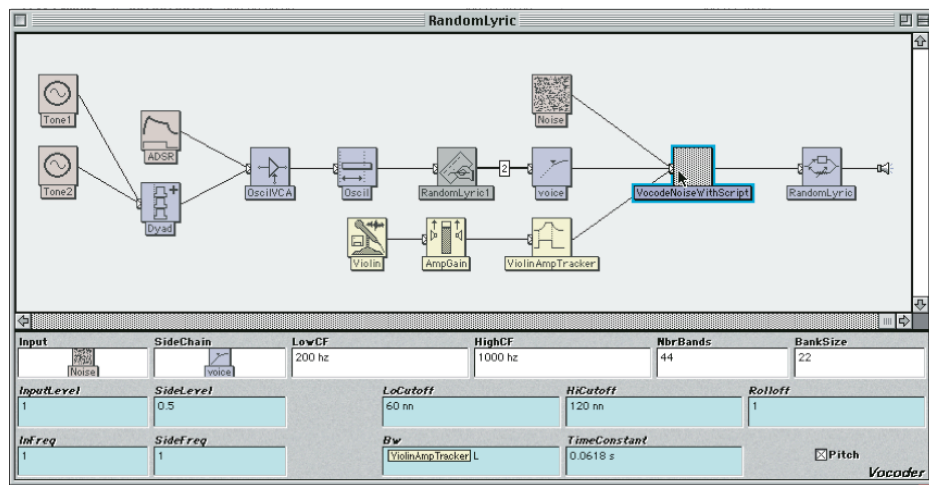


Figure 4. A background lyric layer is stochastically generated in *Lyra* using the composer's program COMP2 (located within the RandomLyric Script Sound). The violin amplitude is tracked and that value is used to control the bandwidth of the Vocoder superSound.

The violin signal is also analyzed for its separate frequency and amplitude data, which are used for direct resynthesis and also as indirect control for other processing algorithms (Figure 5).
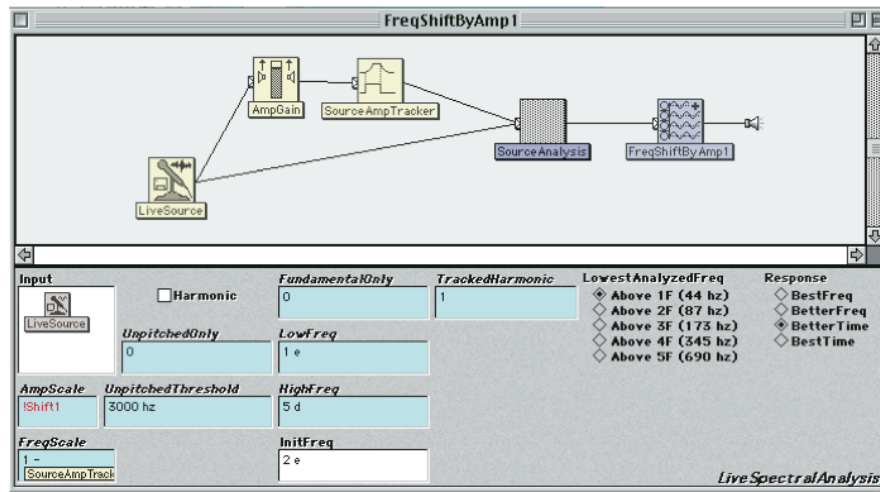


Figure 5. The violin audio signal is analyzed within a LiveSpectralAnalysis Sound. The violin amplitude is also separately tracked and used as an inverse function for the analysis frequency scale. These data are used to resynthesize a new violin layer, whose frequency is inversely modulated by the violin music's amplitude.

The end result is that all of the computer music is generated in real time performance using the live violin music as input data to generate or control parameters within Kyma Sounds (the one exception, the stochastic background layer, does not rely on the violin music but is still generated in real performance time). Many Sounds also employ directed random aspects (e.g., the stochastic background layer as well as several of the interactive Sounds), which are initiated with a seed number that the performer enters into Kyma when the program is compiled prior to performance. The variability allowed the performer coupled with the directed random programming guarantee that the computer sounds will be different with each performance.

## 9. PERFORMANCE ISSUES

For both compositions the performer is placed in a leadership position of starting (and restarting) the computer processing clock time and by providing constant input data for computer processing during performance. The computer music layers are generated by analyzing, resynthesizing, and processing the performer's music during performance time. The analyzed frequency and amplitude data are used for immediate and delayed resynthesis with various parametric adjustments, and are also used as parameter data for other Sounds. Using this

analyzed data for both direct and indirect synthesis as well as for independent processing parameter control creates new compositional possibilities with their own associated aesthetic issues. Technically and aesthetically the past physical distinctions between sound event and phrase, phrase and structure, instrument and performance environment, and composer and performer blur into a series of continuum sets. As Carla Scaletti discovered while designing Kyma: 'Synthesis is just one specific case of analysis' (Scaletti 2002: 74). Small scale events are generated directly with this live performer data, and gestural nuance is used to control middle level phrasing, articulation, and pacing of the computer music. Each informs the other, and this conceptual interaction creates more questions than it answers, which suggests even more future compositions to address these issues.

The computer music also affects the performer. The live music is processed as a direct audio signal through various algorithms, including granulation, ring modulation, frequency modulation, multiple delays, nonlinear wave shaping distortion, reverb, and frequency shifts. The performer is invited to respond to the computer music. This response involves some degree of improvisation, and the composer needs to set technical and aesthetic definitions and boundaries for this improvisation process. The score for *Still Harmless [BASS]ically* invites performer improvisation on phrase and gestural levels, with an open-ended first introduction that is completely improvised. *Lyra* takes this further by inviting the performer to determine the performance order of individual gestures within collected groups of gestures (including repetition and omission) in addition to event level improvisation (embellishment of and deviation from the score). Due to the dynamic real-time analysis and resynthesis capabilities of the Kyma environment the performer determines the path of the composition. 'The mapping is the message' (Scaletti 2002: 79). This mutual performance interaction creates variability on the micro level (and also on the macro level with *Lyra*) from performance to performance, which permits the composition to evolve over time as a personal vehicle for a given performer. The computer becomes an extension of the human performer, creating a 'cyber performance' (Garnett 2001: 30). There is no one definitive or best performance path. And a fixed recording presents only an isolated aural snapshot of one version of the composition.

## 10. CONCLUSIONS
*Still Harmless [BASS]ically* was composed in 2000 to serve as a dedicated testing environment for interactive performance issues. *Lyra* was composed in 2002 to fully integrate the human performer and computer processing on multiple levels. Through a combination of efficient and composer-friendly software code and powerful DSP hardware this experiment has proven successful. Since completing *Lyra* the composer has completed a new composition for piano and Kyma. *(Disturbed) Radiance* (2003) extends this experiment further, exploring more ways to integrate the human performer and the computer processing in performance.

Throughout the twentieth century all aspects of technology experienced an increased pace of development and implementation, and this accelerando will likely continue during the twenty-first century. As technical capabilities change more possibilities are created for composers and performers alike. It is the task of artists to create aesthetically valid and artistically meaningful environments, compositions, and performances using these ever changing tools. 'As software increases in sophistication and hardware becomes more liberally embedded, [the performer] will feel less like … working with a computer than just simply working' (Garnett 2001: 25). Composers, performers, and listeners have the added task of discarding their inherited machine aesthetics from the last century, both the overly positive and the generalized negative. Current directions in computer technology are integrating the various aspects of music creation and performance into a seamless synergy, and there is no reason to assume that this trend can or should be reversed. 'The distinctions we now draw between ourselves and our machines will be blurred and ultimately erased' (Scaletti 2002: 79).

## ACKNOWLEDGEMENTS

## SOUND EXAMPLES
1. Second introductory section of *Still Harmless [BASS]ically*, with computer processing initiated with a WaitUntilFreq Sound that waits for the bass B4 harmonic (488.89 hz). Three Kyma algorithms simultaneously use the bass amplitude as control input data: granulation grain size and grain duration, harmonic resonator frequency, and granulated harmony feedback. (20 s)
2. Beginning of *Lyra*, with computer processing initiated with a WaitUntilAmp Sound. Kyma algorithms include five-part echo feedback, wave shaper, granulation cloud, and random lyric Sounds all using live violin input for real-time processing. (16 s)
3. Middle section of *Lyra* with random lyric algorithm whose vocoder bandwidth is controlled by the violin amplitude. (16 s)
4. Ending *Lyra* gesture. The beginning algorithms are used again along with a double random loop playback, ring modulation (violin – violin product), and a resynthesized violin cloud whose frequency is inversely modulated by the violin amplitude. (42 s)

**REFERENCES**

Belet, B., 1991. Proportional Recursive Stochastic Composition Using COMP2, a Smalltalk-80 Composition Program Within the Kyma Digital Synthesis System. *Proceedings of the 1991 International Computer Music Conference*. International Computer Music Association, pp. 513–16.

Belet, B., 1992. Toward a Unification of Algorithmic Composition, Real-time Software Synthesis, and Live Performance Interaction. *Proceedings of the 1992 International Computer Music Conference*. International Computer Music Association, pp. 158-161.

Garnett, G., 2001. The Aesthetics of Interactive Computer Music. *Computer Music Journal* 25(1): 21-33.

Scaletti, C., 1989. Composing Sound Objects in Kyma. *Perspectives of New Music* 27(1): 42-69.

Scaletti, C., 2000. *Kyma.5 Walkthrough: A Tutorial Introduction to Kyma.5*. Symbolic Sound Corporation.

Scaletti, C., 2002. Computer Music Languages, Kyma, and the Future. *Computer Music Journal* 26 (4): 69-82.

Stockhausen, K., 1996. Electroacoustic Performance Practice. *Perspectives of New Music* 34(1): 74-105.

Truax, B. 2000. The Aesthetics of Computer Music: a Questionable Concept Reconsidered. *Organized Sound* 5(3): 119-126.