Iterated Nonlinear Functions as a Sound-Generating Engine

ARTIST'S ARTICLE

Iterated Nonlinear Functions as a Sound-Generating Engine

Agostino Di Scipio

ABSTRACT

This article describes a method of digital sound synthesis and algorithmic composition based on iterated nonlinear functions. The mathematical framework of Functional Iteration Synthesis (FIS) is outlined, and the dynamics of a specific FIS model are explained. Given the model's peculiarly chaotic system dynamics, an empirical, exploratory attitude is needed in order to achieve compositionally relevant controls. The required method of this exploration is interactive computer music systems. Some examples are discussed bearing on the author's compositional experience with his *Sound & Fury* project. The approach is described in terms of real-time interaction with a source of chaotic but structured flow of sonic information. The relevance of an "ecological" view of composing is emphasized.

FIRST ITERATION

There's no knowledge without chaos. There's no progress without a frequent leave from reason.

—Paul Feyerabend [1]

Musicians working with computers usually refer to two separate areas of concern, namely the creation of large-scale musical structures (computer-assisted composition, algorithmic music) and the creation of the sounds themselves (sound synthesis and/or processing). When I began my efforts in the computer-music field, around 1985, both areas were very attractive for me. However, of primary interest was the merging of them, i.e. the blurring of the clear-cut distinction between the macro-level articulation of musical structure and the micro-level, timbral properties of sound. In other words, algorithmic composing was to result not so much in a music of notes (the "lattice" structure of quantized pitch, duration and intensity values) as in sound textures and complex sonic gestures defined compositionally by their timbre and internal development. Such was my method of conceiving "timbre," here understood as the emergent sonic morphology, as "musical form" itself, and ultimately the very object of composing [2,3]. By the late 1980s I realized that simple mathematical methods such as those found in chaos theory (the modeling of nonlinear dynamical systems) could allow me to pursue the more holistic approach I found necessary. The idea was that both the micro- and the macro-level of music would emerge from a hidden, low-level chaotic dynamics. This eventually opened questions about the "ecology" of compositional processes, extending from the relationship of the sound work to the environment (place or venue) hosting its actual manifestation, to the synthesis of algorithms by which the emergent properties of the overall work were brought forth [4].

At the time when I started this research, musical applications of iterated nonlinear functions had already been proposed as a resource for the computer generation of musical phrases and larger sections [5]. At the 1990 International Computer Music Conference in Glasgow, I had the opportunity to describe my own experiments with this method, and I also illustrated musical examples where iterated nonlinear functions were utilized as a front-end processor for the digital sound synthesis technique known as *granular synthesis* [6,7]. On the same occasion, composer Barry Truax described a similar method developed independently of mine [8]. Then, at the 1991 Colloquium on Musical Informatics in Genoa, I introduced the idea that sound could be generated directly by sampling the trajectory of the *n*th iterate of some nonlinear function [9]. This direct synthesis technique eventually developed into what I call *functional iteration synthesis* [10]. It can be seen as a "non-standard" method for generating sounds with a computer, i.e. as a kind of sound synthesis approach not based on known acoustical models or theories, much in line with the approach of experimental electronic music stimulated in the 1960s and 1970s by composers Iannis Xenakis, Herbert Brün and Gottfried M. Koenig.

Here I would like to summarize the mathematics of functional iteration synthesis (FIS) [11] and discuss issues in the implementation and exploration of an actual working model. Also, I will illustrate examples drawn from my compositional efforts, especially as pursued in contexts of interactive computer music.

THE MATHEMATICS OF FUNCTIONAL ITERATION SYNTHESIS

Let's consider these definitions:

$A \subset \Re$ the set of "init values" for our iterated process;
$G \subset \Re^m$ the set of parameters of the particular map(s) considered;
$B \subset \Re$ the set of samples of the sound signal finally generated;

and the following Cartesian product:

$$A \times G \subset \Re \times \Re^m$$

Let F be a map function defined as

$$F: \quad A \times G \rightarrow B$$
$$(x, \{a_i\}) \rightarrow F(x; \{a_i\}) \text{ where } (\{a_i\} \equiv a_1, a_2, \ldots, a_m).$$

This is a parameter-dependent function, which maps from A to B with a_i as a set of time-changing parameters. Fixing a set of *m* real parameters (a point in G) yields:

$$f: \ A \rightarrow B$$
$$x \rightarrow f(x)$$
$$f(x) \equiv F(x; a_1 \ldots a_m).$$

Provided that $B \subset A$, we can construct a process of "iteration" by repeatedly applying *f* to itself for *n* times:

$$f^n(x) \equiv F(f(\ \ldots f(x) \ldots)) \equiv (f \circ f \circ \ldots \circ f)\ (x)$$

Agostino Di Scipio (composer, music theorist), Via Salaria Antica Est 33/a 67100, L'Aquila, Italy. E-mail: <discipio@tin.it>.

Based on a paper presented at First Iteration, a conference on generative systems in the electronic arts, 1–3 December 1999, at Monash University in Melbourne, Australia.

(in math notation the symbol "∘" denotes "composition," so here we have "n compositions of f as a function of x"). Finally, considering g_i as a sequence of points in G, and provided that $x_{0,i} \in$ A and $g_i \in$ G, we get a sequence of maps f_i. Therefore, a discrete time series is obtained where each sample is the nth iterate of the same function using different parameters:

$$x_{n,i} = f^n{}_i(x_{0,i}) = f_i(f_i(\ldots(f_i(x_{0,i}))\ \ldots))$$

We take this time series as our output audio signal. In other words, the sample stream is calculated as follows:

$$\begin{array}{ll} & \cdot \\ & \cdot \\ & x_{n,i-1} = f^n{}_{i-1}(x_{0,i-1}) \\ & \qquad = f_{i-1}(f_{i-1}(\ldots(f_{i-1}(x_{0,i-1}))\ \ldots)) \\ \text{time} \downarrow & x_{n,i} \quad = f^n{}_i(x_{0,i}) = f_i(f_i(\ldots(f_i(x_{0,i}))\ \ldots)) \\ & x_{n,i+1} = f^n{}_{i+1}(x_{0,i+1}) \\ & \qquad = f_{i+1}(f_{i+1}(\ldots(f_{i+1}(x_{0,i+1}))\ \ldots)) \\ & \cdot \\ & \cdot \end{array}$$

In order to obtain an oscillating signal, any well-chosen nonlinear f can be adopted. Indeed, functional iteration synthesis is less a particular synthesis method than a class of methods sharing the same iterated process as the basic strategy. Adopting a particular function is like choosing one FIS model among many.

The next section illustrates a particular model that I often use in my composing. Clearly, each model has its own characteristics and a peculiar sonic behavior. However, I should stress that the crucial point here is more with the process of "iteration" than the function itself. As was observed some 20 years ago:

> Precisely because the same operation is reapplied . . . self-consistent patterns might emerge where the consistency is determined by the key notion of iteration and *not* by the particular function performing the iterates [12].

ITERATING THE SINE MAP

Consider a monoparametric map (m = 1) such as the classical sine function. The process is described as follows:

$$\begin{array}{lll} \text{F:}[-\pi/2, \pi/2] \times [0, 4] & \rightarrow & (-1, 1) \\ \qquad (x, r) & \rightarrow & \sin(rx) \end{array}$$

The explicit, discrete form of the sine map is very often found in the scientific and popular literature on chaos theory and represents quite a simple algorithm to implement with a computer:

$$x_{n,i} = \sin(r_i x_{n-1,\,i})$$

If you look back at the mathematical framework outlined in the previous section, you will notice that here we have A = $[-\pi/2, \pi/2]$. Indeed, due to the periodicity of the sine function, larger intervals for the init value only return trajectories that are generated starting from within the interval $[-\pi/2, \pi/2]$. The first iterate falls in the interval $[-1, 1]$, completely covered by $sin(rx)$ for x_0 in $[-\pi/2, \pi/2]$ and r equal to or greater than 1. Notice, also, that we have G = $[0, 4]$. This is because any larger r value would provide results quite similar to, if not perfectly identical with, those provided with r just smaller than 4.

The effective behavior of this FIS model could be graphically rendered by tracing the nth iterate of some x_0 while linearly increasing r. Indeed, that is the method usually adopted to trace the well-known "bifurcation diagram" (characterizing the dynamics proper to nonlinear systems) [13]. As is expected, a higher iterate yields more active and dynamical trajectories. To my knowledge, a mathematical description of these oscillating trajectories, especially as relative to the numerical relationship among the trajectories and their derivatives, has not been specifically addressed in any scientific study. Even if it has, it is still most likely that further insight into the mathematics of functional iteration synthesis might reveal details of chaotic nonlinear dynamics as yet unexplored. As a matter of fact, we find ourselves in a territory where little help is provided by purely analytical means, and we can profitably turn to a more exploratory, experimental methodology.

When a higher iterate is calculated, the init value x_0 is soon forgotten: one cannot tell where in the interval $[-\pi/2, \pi/2]$ the iteration process was started. Transients disappear and the final plot for the process becomes coincident with the image similar to the popular "bifurcation diagram" (Fig. 1).

EXPLORING THE PHASE SPACE

The output sound of the FIS model just outlined is dependent on the particular orbit traced by the parameters in the process phase space, i.e. in the area $[-\pi/2, \pi/2] \times [0, 4]$. In general, given some monoparametric map, the orbit is defined as the coupling of the two series r_i and $x_{0,i}$. The time series $x_{n,i}$ (i.e. our digital audio signal) is the resultant of the coupling, and its detailed waveform shape is also determined by the orbital velocity across the phase space (Fig. 2). For a qualitative description of the process dynamics, let us consider now the three parameters r, x_0 and n, separately.

Broadly speaking, the scaling factor r determines the kind of behavior, or waveform shapes, in the output sound signal, ranging from very gentle curves (e.g. r = 2) to wild and highly intricate oscillations (e.g. r = 4). Many regions are included in between, either featuring multi-periodic cycles (limit-cycle attractors), or aperiodic oscillations (strange attractors). For digital signal processing practitioners, the latter can be reminiscent of frequency- or phase-modulations.

The init value x_0 determines the actual sample values. Slightly different init values may result in sound signals that, while almost identical at the outset, gradually shift apart from one another as the process continues. Here is a typical sign of "dependency on the initial conditions," a symptom of chaotic dynamics.

Finally, the number of iterations, n, contributes to the spectrum bandwidth in the output sound. With larger values, the sound signal goes through wider and wider oscillations, and the spectrum gets denser and richer in higher frequencies. Eventually, broadband noise is obtained. In most cases, n = 9 already returns (almost white) noise, regardless of the particular values for r and x_0.

At synthesis run time, we might want to (1) change r and keep x_0 constant; (2) change x_0 and keep r constant; or (3) change both r and x_0 (the number of iterations, n, cannot change during computation, as that would most probably cause undesired clicks in the sound signal). These controls determine, respectively, (1) quick changes in the signal waveform (highly dynamical spectra); (2) different signal waveforms of the same *kind*, i.e. sharing similar global properties as captured by r (constant bandwidth spectra); and (3) a mixture of the two, often heard as articulated sound textures in a continuous flux of change, maybe including sudden "pauses" (sub-audio frequencies, or even DC offset). Notice that, in general, the actual implementation of the model must include the possibility of varying either r or x_0 through time—otherwise there will be no output sound at all!

By driving either r or x_0 with periodic functions, we can determine closed orbits in the phase space, such that the synthesis process is forced to periodic behavior. Accordingly, the sound output will include an audible patterning of sound shapes, either in the sub-audio (rhythms) or audio range (pitches).

The time-changing relationship between r_i and $x_{0,i}$ is crucial with respect to the morphology of the generated sound.

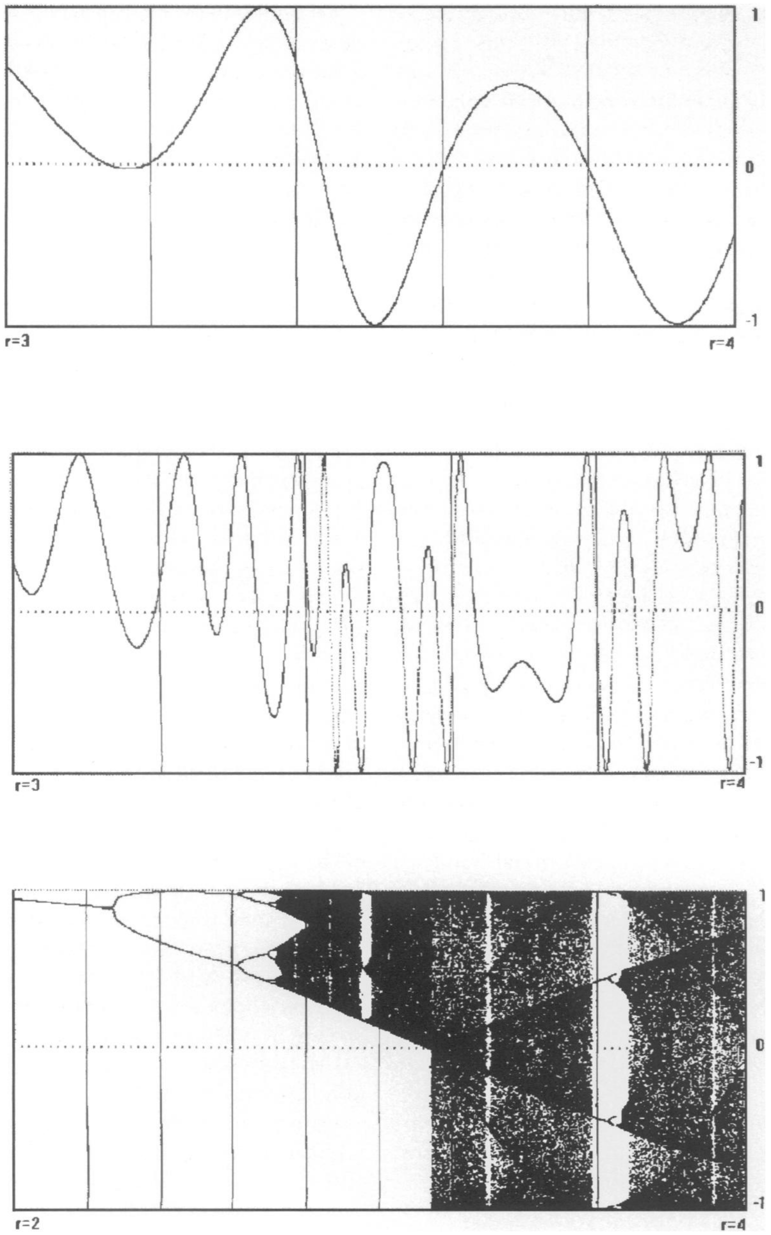
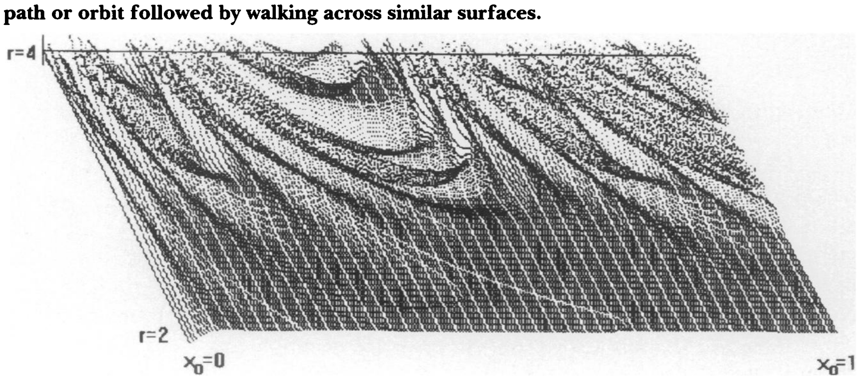


Fig. 1. Plotted from top to bottom are the 5th, 7th and 100th iterations of the sine-map function (the latter represents the "bifurcation diagram" proper to the particular iterated non-linear function).

Fig. 2. Surface of the sampled phase-space region [0,1] × [2,4] of the iterated sine function, as relative to the 5th iterate. The output audio signal of FIS models corresponds to the path or orbit followed by walking across similar surfaces.



Even more crucial is the fact that it yields different results when a different iterate, n, is considered, i.e. it varies with the considered iterate. In a typical situation, one cannot modify any of these three values without also causing, as a side effect, a change in the way the other two affect the overall result. In short, we are left only with the possibility of a qualitative characterization of the interdependency among parameters, as opposed to a quantitative, analytical characterization. In a sense, this phenomenon reflects the "non-integrability" of mathematical models of dynamic systems. FIS reveals the complexity of chaotic systems to the ear.

Now, far from being a source of frustration and an obstacle for the composition process, I think this situation fosters an exploratory attitude that in the end may lead a composer or sound designer towards a fresh and renewed perspective on the nature of the sound material. Using iterated nonlinear functions as a sound generating engine, one has to learn the sonorous possibilities and the musicality hidden in the process and finally liberate its aesthetic potential. The most appropriate strategy becomes one of an empirical investigation of the parameters in play. The exploration of the phase space turns into an empirical exploration of the network of parameter interconnections. It becomes crucial, then, to turn to interactive computer music systems for a more sensible understanding of the model dynamics.

ISSUES IN THE REAL-TIME IMPLEMENTATION OF FIS

For a useful interactive implementation of the sine map model of FIS [14], some basic prerequisites should be matched. It should be possible to update r and x_0 at sample rate, each sample being the nth re-application of f to x_0. The sample loop should be of variable size (variable timing), to allow any number of iterations n to be specified. As a consequence, ad hoc tricks are necessary in order to synchronize the sample loop, with other operations taking place at sample time.

Provided these conditions are fulfilled, one can implement the sine map model of FIS in a relatively straightforward way, maybe only needing to rewrite the explicit discrete form in some programming language. In general, two nested loops are required, the sample loop and the iteration subloop. I did this with Symbolic Sound Corp.'s KYMA™ sound-design workstation, manipulating its object icons in a graphical user interface. As an alternative implementation, I wrote a short piece of Motorola 56002 micro-code and added it to the KYMA system micro-sound library.

Another interesting approach exists, in which FIS is understood as a kind of generalized *waveshaping* synthesis (WS). In classical waveshaping [15], a series of samples (input signal) is mapped onto a given numerical interval by a particular transfer function, called the *waveshaper* (standard transfer functions are calculated by summation of Chebichev polynomials). This is a typical mapping operation. In order to implement the iteration, we just iterate the mapping: the output of a single transfer operation is fed back into the process and used as a new input value, i.e. as a new value to map rather than as the audio sample, as would be the case in normal waveshaping. The mapping is then re-applied as many times as required. Only the result of the last mapping (i.e. the nth iterate) is finally taken as the audio sample [16].

It can be demonstrated that, with a particular transfer function, iterated WS is made identical with the sine map model of FIS [17]. Indeed, in retrospect classical WS can be viewed as a special case of the broader theoretical framework of FIS; for WS, we set $n = 1$ and take some Chebichev polynomial summation as the nonlinear function. There is, however, a difference between the two approaches, which I consider relevant both in theory and in practice. With WS, in principle one achieves complex sounds by starting with the periodicity inherent to the waveshaping process and augmenting the system's complexity. With FIS, instead, one starts directly within a peculiar situation of chaos, heard as a kind of sonic turbulence, and then looks for—and eventually finds—isles of order and emergent periodic patterns (local singularities in the system dynamics).

SOUND & FURY

Iterated nonlinear functions have a relevant part in my *Sound & Fury* cycle of compositions (1995–1998). Each work represents a different manifestation of the very same low-level dynamics in a different medium: included are tape pieces, live computer music concert pieces and sound installations. Eventually, the overall project evolved into a kind of music-theatre, with a libretto based on short excerpts from Shakespeare's *The Tempest* (as well as from W.H. Auden's *The Sea and the Mirror*, a commentary to the *Tempest*, and from Eugenio Tescione's re-reading of Auden's commentary). The full version, premiered in January 2000 at Theatro Garcia de Resende in Evora (Portugal), is scored for two actors, two percussions, interactive computer music system, 8-track tape and multiple slide projections. The introductory work, *Natura allo Specchio* (Italian for "nature standing by the mirror"), is a tape made by mixing various takes of my own live performances with extended FIS algorithms. The work was first played at the festival "Ruido," in Mexico City, then at the First Iteration Conference concert, at Monash University, Melbourne, December 1999 [18]. It can be heard as a kind of sonic environment, with myriads of little sounds and noisy droplets, a mysterious auditory scene very lively and almost "natural" to the ear and yet utterly synthetic in nature. A voice is also featured (thanks to Simon Emmerson for lending his), which speaks a *Tempest* fragment that I find very appropriate to set forth the conceptual and aesthetic stance for the *Sound & Fury* project, albeit in a rather self-mocking way:

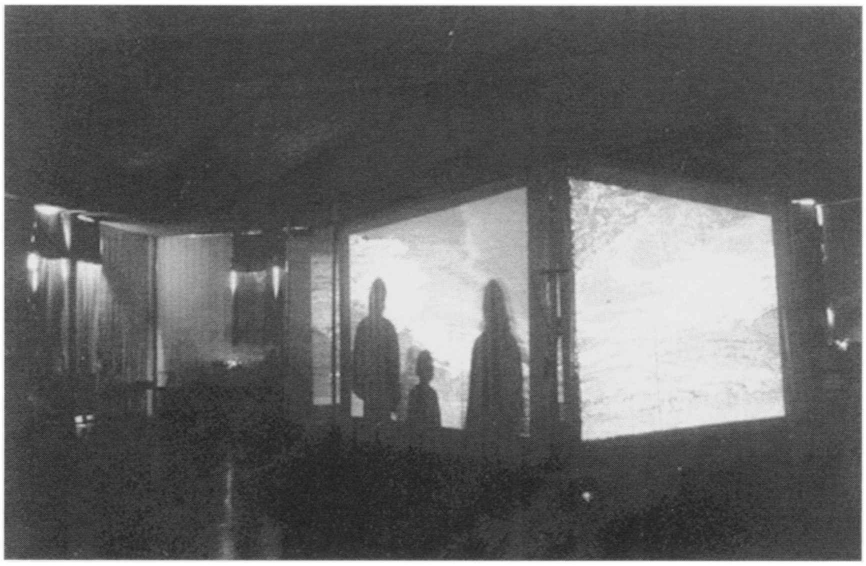
> Hang you whoreson, insolent noise maker!
>
> —Shakespeare [19]

The last work in the cycle is an audio-video installation, with computer-controlled slide projection, pre-recorded tapes and interactive computer music (Fig. 3). In the next section, I discuss some practical aspects of "interactivity" as implemented for this work. First I want to outline the overall constructive schema captured in the computer.

As the unifying strategy of the overall project, the *Sound & Fury* works involve (1) the sine-map model of FIS as the primary sound engine component (micro-level, sound synthesis engine), as well as (2) more iterated functions as automated music generators (macro-level, algorithmic composition). The latter represent a real-time computer "score" that calls for several instances of the former and initializes them with appropriate parameter values. The orchestra/score paradigm, however, only partially applies here, as the score's generation itself is made dependent, in real time, on actual features in the generated sound stream (see below). The macro-level functions are very similar to, if not identical with, the micro-level ones. Therefore, the core of the compositional process represents a kind of iterated function system (IFS) distributed over two distinct levels. At the beginning of a new performance, the init values of the macro-level iterated process are reset, causing timbre nuances and other relevant musical dimensions, such as the pace and density of events, to change accordingly, although in unpredictable ways. Therefore, every performance has its own internal variations.

On top of that autonomous "music machine," or automated music component, in the *Sound & Fury* project all functional iteration processes are controlled in real-time during performance (this applies also to the tape pieces, where composition took place by recording and pasting together several performances). For the sake of brevity, I will refer to two distinct interaction *lay-*

Fig. 3. A snapshot of the interactive installation *Interactive Island (Sea Lights and Colors)*, Nettuno, Italy, March 1996. (Photo: Manilio Prignano)



ers: (1) human/machine and (2) machine/ambience. If I deem it relevant to review them here, it is because the decisions I had to make concerning interactive control processes were often influenced by the peculiar complexity arising from the parameter interdependency in the iterated functions.

Human/Machine Interaction Layer

The first interaction layer is controlled by a simple virtual control surface (VCS) that I devised for myself in Kyma. Every virtual fader in the VCS represents a manually controllable MIDI fader. Fader values are mapped onto various parameter values, including the cycle time for x_0—the only cue to the orbital velocity in the FIS phase space (this can eventually be used as a kind of "pitch wheel," when orbital velocity reaches the audio frequency range; x_0 values used in actual composition range from a few seconds to thousandths of a second).

The r parameter and the particular iterate, n, are not controlled in the interactive VCS. Actual values of r are left to the internal automated music component and are composed in the interval [20]. They change with time according to some pre-calculated envelope shape, such that the spectrum bandwidth varies. The amount of variation, however, is determined by another VCS fader.

Volume faders are provided in the VCS to make it possible for the performer to compensate for different amplitude levels emerging from the random combinations in the process parameters (amplitude may change very radically, depending on the particular phase space region being visited).

In such a situation, the composer or performer handles a chaotic and yet structured flow of sonic information arising from the machine. Using terminology drawn from recent music theory [21], this can be described as a kind of *interpretational design*, reflecting a mode of computer-assisted composition and performance where the task is one of interpreting and making sense of the data generated by the computer, possibly in the form of a list of alphanumeric symbols. A classical example in computer music is Koenig's Project One, dating from the early 1960s [22]. However, in the present case, interpretation is operated in real-time, based on immediate perception of an articulated sound stream, made of changing sonic gestures and textures. So a more appropriate definition would be *interactive interpretational design*.

In some *Sound & Fury* works (namely the second and fourth), human/machine interaction extends to one or more percussionists playing on stage. As they are playing, their sound level is tracked by the computer and properly scaled in order to modify the cycle time values for x_0 (simple amplitude followers are used, but the mapping from amplitude to orbital velocity—hence to frequency and/or rhythm—is far from trivial). Furthermore, with some of the percussion instruments, as the live sound gets louder, the FIS parameter orbit gets quicker, such that the rate of timbre and pitch change in the synthesis becomes quicker. With other percussions, the inverse applies. Finally, the percussion level is also utilized as a control signal to determine tempo changes in the automated music component: the louder the percussion sounds, the slower the timing of events (variable scheduling is a relevant feature of the Kyma software). This is useful to compensate for the density and intensity in the overall sound fabric at any moment during the performance and ensures that the interaction never reaches "catastrophic" points of non-return (systemic saturation).

Machine/Ambience Interaction Layer

The second interaction layer plays a relevant role in all of the live performed works of the *Sound & Fury* project, most notably in the last work, the interactive sound installation *Interactive Island (Sea Lights and Colors)*. As real-time FIS sounds are generated and played all over the performance space (or any other "ambience," to put it more generally), they are also captured by a number of microphones and sent back to the computer. The computer analyzes the numerical difference between the waveform of the fed-back signal and that of the synthesis signal. The difference tracks the timbre modifications resulting from amplification and room acoustics (including the presence of visitors, their relative position in the installation space, their actual number). The difference values are used as a control signal to modify the FIS parameters and the timing in the low-level automated music component. In short, both the "dead," abstract data structure captured in the computer and the live room acoustics of the material ambience become responsible for the development of the musical flow, in terms of fluctuations in various musical dimensions. The shape, material and history of the particular ambience are thus projected into the timbre and the articulation of the sound stream.

Ultimately, the design strategy here is one of implementing the sonic concept in the form of a whole system capable of self-organization by means of a permanent exchange of information with the surrounding environment. While it is clear that further efforts are needed in this direction, the *Sound & Fury* project at least led me to understand the network of connections out of which interactive behavior is shaped as an object itself of composing [23]. The sound work, then, is seen as the audible epiphenomenon of a kind of *eco-system*. (Maybe it is not by chance that some of the most peculiar sonorities created with interactive FIS models are reminiscent of environmental sonic textures and other natural sound events [24].) Understanding interaction as the object of composition means that the internal ecology of the musical process is captured in the mutual, causal interconnection of many component elements: changes in the ambience response (caused, for example, by visitors walking around the installation or speaking among themselves) determine unpredictable but consistent reactions and adaptations in the machine's behavior (and among its internal process components), which in turn causes unpredictable but consistent reactions and adaptations in the ambience and the visitors' behavior.

By Way of Conclusion

The aim of this paper was to review the method of functional iteration synthesis and to discuss its implications for real compositional experience. The intricate interdependency among its parameters suggests that musical applications of FIS are subject to open-ended explorations, as made practicable by interactive systems. This is made very clear already by the audible manifestations of the FIS process, as in fact the sound of the simplest implementation is heard as low-frequency turbulence with sudden transition to broader band noise. In the end, based on the compositional experience just outlined, the need for a careful study of various modes of interaction emerged from the very nature of notions such as *iteration* and *nonlinearity*—i.e. from the concept itself of *deterministic chaos*.

For me, the idea of interacting with a chaotic but structured source of infor-

FIRST ITERATION

mation is artistically fruitful in that it leads to a perspective where the musician deals with the emergent behavior or epiphenomenon of a network of process components, whose ecology becomes a structural element of composing. Using functional iterations as a sound-generating engine in interactive contexts, the emergent sound texture reveals audible traces not only of the underlying nonlinear model, but also of the historical and material place where listeners gather together to be part of the music they listen to [25]. However, the medium of sound installation is by no means the only medium in which it makes sense to speak of the ecology proper to the process initiated by the composer.

It is difficult here to draw any "conclusions," albeit tentative or preliminary ones. I think this kind of artistic and intellectual commitment, at the interface between music and science, opens a vast enough territory as to make it hard to see where and when the search will eventually reach an end—or even what will be the next step. Using iterated nonlinear functions supports a mode of art making and an aesthetic experience by which an artist learns his/her own path and orientation only by exposing him/herself to the actual, perhaps chaotic sound texture and makes sense of the bits of order that eventually emerge from it. It is like manipulating and putting into context a mechanism or source of hidden structured information in sound: developments remain largely unpredictable and literally unheard before any real interaction takes place between an observer (such as a composer) and the mechanism's autonomous manifestation—or between the mechanism and its surroundings.

This creative scenario fits very well with a notion that musical thinking never exists without a relatively satisfying empirical understanding of its available means or tools (there is nothing like an "autonomous" music or musical idea). It also fits with the notion according to which "art is self-alteration" (e.g. John Cage). By making the technology of this art (the computer) a technology for "self-alteration" rather than a means for "world a(du)lteration," we may help to turn a technology of power and control (today a technology of *communication* power and control) into a means of cultural freedom and diversity.

> Music, in its final appearance, preserves at least traces of the processes by which it emerged from chaos.
>
> —Herbert Brün [26]

References and Notes

1. Paul K. Feyerabend, *Against Method. Outline of an Anarchistic Theory of Knowledge* (London: New Left Books, 1975) p. 146.

2. Agostino Di Scipio, "Micro-Time Sonic Design and the Formation of Timbre," *Contemporary Music Review* **10**, No. 2, 135–148 (1994); and Agostino Di Scipio, "Formal Processes of Timbre Composition Challenging the Dualistic Paradigm of Electroacoustic and Computer Music," *Proceedings of the International Symposium on Electronic Arts* (Helsinki, Finland: 1994); also available on-line at <http://www.uiah.fi/bookshop/isea_proc/nextgen/j/19.html> and reprinted in *Proceedings of the International Computer Music Conference* (Aarhus, Denmark: 1994).

3. Agostino Di Scipio, "Inseparable Models of Material and of Musical Design in Electroacoustic and Computer Music," *Journal of New Music Research* **24**, No. 1, 34–50 (1995).

4. Agostino Di Scipio, "Synthesis of Environmental Sound Textures by Iterated Nonlinear Functions," paper presented at the *2nd COST G-6 Workshop on Digital Audio Effects* (Trondheim, Norway: 1999); to appear in *Proceedings of the 13th Colloquium on Musical Informatics* (L'Aquila, Italy: 2000).

5. See, for example, Jeff Pressing, "Nonlinear Maps as Generators of Musical Design," *Computer Music Journal* **12**, No. 2, 35–46 (1988).

6. Agostino Di Scipio, "Composition by Exploration of Nonlinear Dynamical Systems," *Proceedings of the 1990 International Computer Music Conference* (Glasgow, Scotland: 1990). An implementation of the method is described in Agostino Di Scipio and Graziano Tisato, "Granular Synthesis with the Interactive Computer Music System," *Proceedings of the 10th Colloquium on Musical Informatics* (Milan, Italy: 1993). I used these granular methods in a number of compositions, including *5 piccoli ritmi*, for taped guitar sounds and interactive computer processing (available on Neuma Rec., 450–99).

7. For a broader view on granular synthesis, please refer to Curtis Roads, "Asynchronous Granular Synthesis," in Giovanni De Poli, Aldo Piccialli and Curtis Roads, eds., *Representations of Musical Signals* (Cambridge, MA: MIT Press, 1991) pp. 143–186; Barry Truax, "Real-Time Granular Synthesis with a Digital Signal Processing Computer," *Computer Music Journal* **12**, No. 2, 14–26 (1988). For an ecological view of granular synthesis, see Damián Keller and Barry Truax, "Ecologically-Based Granular Synthesis," *Proceedings of the International Computer Music Conference* (Ann Arbor, Michigan, 1998), also available on-line at <http://www.sfu.ca/~dkeller/EcoGranSynth/EGSpaper.html>.

8. Barry Truax, "Chaotic Nonlinear Systems and Digital Synthesis. An Exploratory Study," *Proceedings of the 1990 International Computer Music Conference* (Glasgow, Scotland: 1990).

9. Agostino Di Scipio, "Caos deterministico, composizione e sintesi del suono," *Proceedings of the 9th Colloquium on Musical Informatics* (Genoa, Italy: 1991).

10. Agostino Di Scipio and Ignazio Prignano, "Synthesis by Functional Iterations. A Revitalization of Non-Standard Synthesis," *Journal of New Music Research* **25**, No. 1, 31–46 (1996). Sound examples accompanying this paper can be downloaded from <http://www.swets.nl/jnmr/vol25_1.html#discipio25.1>.

11. The acronym "FIS" should not be mistaken for the more popular "IFS," standing for *iterated function systems*. The term *functional iteration* is found in some chaos theory literature (e.g. see [12]).

12. Mitchell Feigenbaum, "Universal Behavior in Nonlinear Systems," *Los Alamos Science* **1**, No. 1, 4–27 (1980). Reprinted in *Physica* **7** (1983) pp. 16–39.

13. Robert May, "Simple Mathematical Models with Very Complicated Dynamics," *Nature* **261** (1977) pp. 469–467; Pierre Collet and Jean-Pierre Eckmann, *Iterated Maps on the Interval as Dynamical Systems* (Boston, MA: Birkhauser, 1980).

14. For a deferred-time iterated sine map implementation, readers are referred to the Csound implementation downloadable at <http://members.es.tripod.de/csound/> (by Josep M. Comajuncosas), as well as to my own lecture, with Csound examples, "Sound Synthesis by Iterated Nonlinear Functions," included in Riccardo Bianchini and Alessandro Cipriani, eds., *Virtual Sound* (Rome: Contempo, 2000), appendix.

15. Daniel Arfib, "Digital Synthesis of Complex Spectra by Means of Multiplication of Non-Linear Distorted Sine Waves," *Journal of the Audio Engineering Society* **27**, No. 10, 757–768 (1979).

16. Before I realized that the mathematics of chaos theory would provide a more appropriate conceptual framework for implementing iterated processes, I had intuitively used iterated waveshaping for sound synthesis, especially to simulate reed instruments (see Agostino Di Scipio, "Distorsione nonlineare con funzioni sinusoidali," paper presentation at the 1987 *Colloquium on Musical Informatics* in Rome). At about the same time, several authors observed that physical modeling of reeds requires the formulation of dynamical systems with chaotic properties. See, for example, Charles Maganza and Robert Caussé, "Bifurcations, Period Doublings and Chaos in Clarinetlike Systems," *Europhysics Letters* **1**, No. 6, 295–302 (1986); Vincent Gibiat, "Chaos in Musical Sounds," *Proceedings of the Institute of Acoustics* **12**, No. 1, 511–518 (1990); Angelo Bernardi, Gian-Paolo Bugna and Giovanni De Poli, "Musical Signal Analysis with Chaos," in C. Roads, S.T. Pope, A. Piccialli and G. De Poli, eds., *Musical Signal Processing* (Lisse, the Netherlands: Swets & Zeitlinger, 1997) pp. 187–220.

17. Agostino Di Scipio, "Kyma Tips: Functional Iteration Synthesis" (manuscript downloaded via file protocol transfer from <ftp.shout.net/pub/symsound/kymasnds/> 1996).

18. Available on compact disc BUG108 <http://www.solitary-sound.com/>.

19. William Shakespeare, *The Tempest*, Act 1, Scene 1, line 43.

20. Di Scipio [3,4].

21. Otto Laske, "Towards an Epistemology of Composition," *Interface—Journal of New Music Research* **20**, No. 3–4, 217–234 (1991).

22. Gottfried M. Koenig, "Project One: A Programme for Musical Composition," *Electronic Music Reports of the Institute of Sonology* **2** (1969).

23. Michael Hamman, "From Symbol to Semiotic: Representation, Signification and the Composition of Music Interaction," *Journal of New Music Research* **28**, No. 2, 90–104 (1999).

24. See Di Scipio [4].

25. Agostino Di Scipio, "Le son dans l'espace, l'espace dans le son," *Doce Notas Preliminares* **2** (1998) pp. 133–157.

26. Herbert Brün, "Infraudibles," in J. Beauchamps and H. von Foerster, eds., *Music by Computer* (New York: John Wiley, 1966), p. 120.

Agostino Di Scipio's compositional output includes computer music, works for soloists interacting with computers, and sound installations. Representative works have appeared on CDs from Neuma (New York), Capston (New York), NoteWork (Cologne), ORF (Vienna) and other labels. His writings, published in international journals and anthologies, are often devoted to the relationship of art to science and technology, and also to the history of music technology.