Composing at the intersection of time and frequency*

MICHAEL CLARKE

Department of Music, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, England

Many software packages for computer music encourage the composer to take either a time domain approach or a frequency domain approach. This paper examines the possibilities afforded by recent software developments of working at the intersection of these two domains. It investigates the relationship between the FOF algorithm, originally used in the CHANT program, and more traditional approaches to granular synthesis, and considers how they can be combined. The author's compositions are used as illustrations of these techniques. The significance of using the FOF algorithm in granulating sound files is explained (FOG). Methods of using and controlling the FOG unit-generator are explained. Compositional and aesthetic issues arising from working with sound at this ambiguous intersection are investigated.

1. TIME AND FREQUENCY

Contrasts between the time and frequency domains are often made in computer music literature and it is easy to think of them as opposites, as exclusive of one another. In reality, however, they are closely interrelated and it is rarely possible to consider either without the other: they are two aspects of the same phenomenon. Changes in either dimension almost invariably have consequences for the other.

The interconnection of the domains is well known to composers working in the electroacoustic field. For example, changing the speed of an analogue tape or the sampling rate of a digital recording on playback changes not only the evolution of the sound in time but also its frequency. More subtly, filtering a sound, performing an operation to change its frequency content, also changes it in the time domain. Sharp attacks and decays of envelopes can be blurred, and the more precise the action of the filter is made in the frequency domain (the narrower the bandwidth or the higher the 'order' of the filter) the greater the effects in the time domain. This is because filter algorithms work by manipulating the waveform, the time domain representation of the sound. To the original signal are added weighted, delayed copies of itself (either recursively or non-recursively) resulting in a blurring of the envelope. A manipulation of the time domain is used to effect a transformation in the frequency domain. There are many other examples of the interaction and indivisibility of the two domains. The domains are more a matter of perceptual focus than a substantive distinction.

Frequently composers find the interrelation between time and frequency a hindrance and algorithms such as the phase vocoder are used to try and disentangle the two domains. Alternatively, it is possible to try to use the interrelation and ambiguity as a deliberate compositional device. This paper describes my interest in such an approach which I have investigated over a number of years, particularly in the context of granular synthesis. It describes how a number of different computer music programs can be used to cross the time/frequency divide and play with the perceptual ambiguity this creates.

2. APPROACHES TO SYNTHESIS

Many traditional synthesis methods encourage composers to focus their attention on the frequency domain. In additive synthesis, for example, the composer specifies the spectrum of a sound in terms of the frequencies and strengths of partials at any particular moment in time. With subtractive synthesis the composer specifies the centre frequencies and bandwidths of filters. One of the challenges that composers frequently find themselves facing in using such methods is how to make the sounds come alive in time. All too often synthesised sounds lack the liveliness of natural sounds. This is because most natural sounds follow a complex timbral evolution through time. Synthesis methods that focus attention on the frequency domain often make it difficult to match this rich morphology. Of course there are ways to counter the problem: individual partials can be given sophisticated amplitude envelopes and subtle pitch envelopes. These controls, however, can prove very time consuming and tedious to specify by hand. More recent developments have resulted in control data being derived from computer analysis of natural sounds, or generated by complex algorithms. The results can be very impressive and have helped to alleviate the prob-

* The work documented in this paper was made possible through the opportunity of working at a number of studios, and with the help of many people. In particular, I would like to thank Peter Manning (University of Durham), Tamas Ungvary (EMS, Stockholm), Xavier Rodet (IRCAM, Paris) and Barry Truax (Simon Fraser University, Vancouver). Thanks also to Paul Archbold (University of Huddersfield) for help in drawing the illustrations.

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

lems of an essentially static, frequency-oriented approach.

Granular synthesis on the other hand starts from the time domain. First proposed theoretically by Gabor (1947) and musically by Xenakis (1971), it was later developed by Roads (1978, 1988, 1991) and Truax (1988, 1994), amongst others. Granular synthesis constructs sounds by putting together numerous tiny grains of sound (usually too short to be heard properly in themselves but audible as part of a stream). Rather than approaching sound as a group of frequencies to be assembled 'vertically', outside time, granular synthesis views sound as a complex pattern of grains to be assembled in time, 'horizontally'. Traditionally in granular synthesis the parameters of the grains are continually changing, even if only by the smallest amounts. Statistical functions control each of the grain's parameters within specified limits. The result is sound that is never static, it is constantly varying, evolving and transforming. In this respect granular synthesis, with its emphasis on the time domain, can offer a welcome alternative to the sometimes rigid and static qualities of frequency-based synthesis methods. However, in gaining in the area of temporal variety, granular synthesis has frequently lost something in terms of spectral control: it has proved less possible to control the resulting spectrum which has often been a side effect of the temporal processes rather than a parameter to be shaped in its own right.

Bringing together the time and frequency approaches to synthesis offers the potential of lively, richly time-varying sounds that nonetheless have carefully shaped spectra. Furthermore, the ambiguity and the transformational possibilities of moving between the time and frequency domains can be used as an expressive compositional device.

3. IMPAC: TIMBRE OUT OF TIME

IMPAC was a real-time granular synthesis program developed at EMS in Stockholm by Michael Hinton in the late 1970s. Like most granular synthesis programs its primary concern was the distribution of grains of sound in time. It was not especially concerned with the frequency domain. A PDP 15 computer[1] provided the user interface and sent control messages to custom-built digital FM and analogue oscillators which generated the sounds. The system is unfortunately no longer operational, a casualty of the rapid advance of technology. When launched, the program automatically generated a sequence of grains with pseudo-random variations to a number of the grain parameters. Some of the most significant user parameters were not given specific values but rather the upper and lower bounds were specified within which the program then selected values at random. The composer's role was, therefore, to shape the mass of randomly generated grains by adjusting the limits of random choice. Control was over the range in which grains were distributed rather than in the determination of particular pitches or amplitudes. The upper and lower boundaries for each parameter could be preset or adjusted in real-time using a number of assignable controllers. These included a joystick and a digitiser pen. The x and y axes of both these controllers could be assigned to particular variables. The composer was therefore able to shape the distribution range of clouds of sound in real-time. The parameters included upper and lower limits for the pitch and amplitude of grains, the density of grains (how many grains per second), the grain duration, the FM modulation frequency and FM ratio.

The underlying orientation of the program towards statistical distribution in the time domain could, however, be subverted. It was possible to restrict the range of the computer's random choice to zero by setting the maximum and minimum values for a particular parameter to the same value. The maximum and minimum values for the pitches of grains, for example, could be assigned to the axes of the joystick in such a way that the composer determined the pitch precisely and the random element of the program was eliminated. Although the pitch would change with the movement of the joystick there was in fact no random range available to the computer (figure 1). This resulted in a sequence of grains all of the same pitch and, with a high density (a large number of grains per second), these would often blend into what was apparently a single sus-

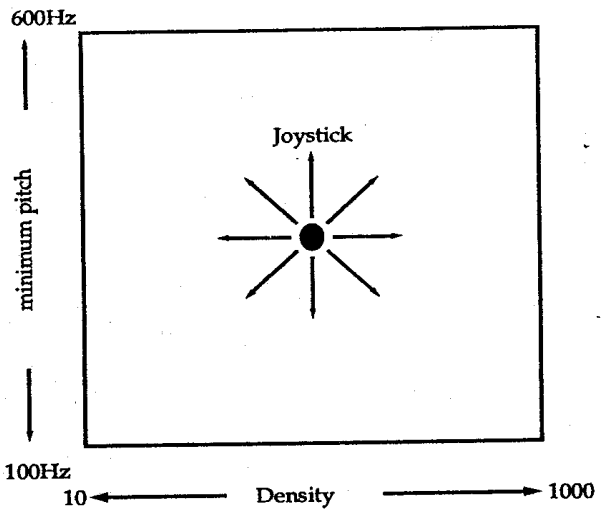


Figure 1. Settings to eliminate random pitch selection.

[1] The PDP 15 was a large and, in its day, fast mainframe computer produced by the Digital Equipment Corporation (DEC). The PDP 11, referred to later in this paper, is a smaller minicomputer also from DEC.

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

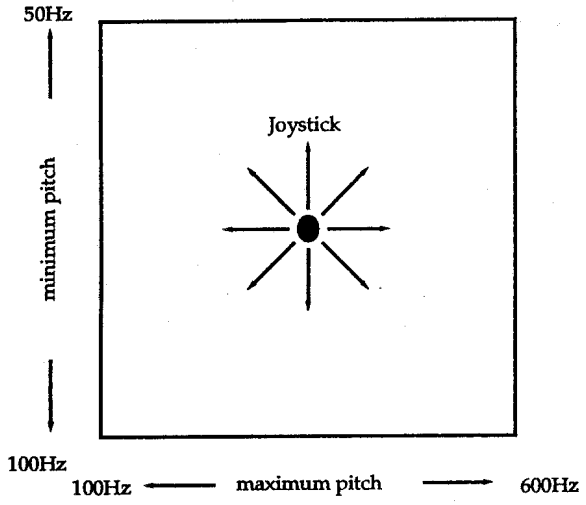


Figure 2. Movement from fixed to random pitch.

tained note (but was in fact a rapid succession of identical grains). However, a more interesting result could be obtained by still leaving a small degree of randomness available and setting the upper and lower limits to slightly different values. The aural result was then a rich and constantly varying sustained timbre, much more lively than many syntheses generated by frequency domain methods.

Other settings of the real-time controllers made it possible to move dynamically between a deterministic situation with the controller in one position (e.g. bottom left-hand corner) and a highly random distribution in another (figure 2). At one extreme the listener's focus would be on the frequency domain, a timbre with a clear pitch, at the other on the time domain, an evolving texture of grains. The movement between these extremes and the ambiguity that at times arose as the centre of focus shifted provided a rich source of compositional material. I explored the compositional potential of these approaches in a work composed in Stockholm, *Uppvaknande* (Clarke 1984). Sound example 1 is taken from this work and illustrates the granular monodies produced using IMPAC.

With IMPAC the manipulation of such transformations was rather restricted. However, the real-time controls enabled composers to work intuitively and to learn the potential of the system through interactive aural experience. The system was intended for composition in the time domain and not surprisingly control over the frequency domain was limited. The division of the control and generation processes between different hardware units (not unlike MIDI control of a synthesizer) limited the amount of detailed shaping of grains that was available. High densities also caused problems as grains would sometimes cut in on each other (similar to a MIDI synthesizer running out of polyphony), a new grain interrupting an existing one to take over its space and in so doing causing distortion which could only be masked by using a significant amount of reverberation. None the less, the general sound quality of the system was very high and its approach to real-time granular synthesis a very imaginative one.

4. FOF: TIME OUT OF TIMBRE

4.1. CHANT

Whereas with IMPAC the user's attention was focused on the time domain and on random choice, CHANT is primarily designed to provide detailed control over the frequency domain. It allows the user to specify with great precision, and yet economy of commands, the spectral envelope of a sound. CHANT was developed at IRCAM by a team led by Xavier Rodet (Rodet, Potard and Barrière 1984). As the name of the program suggests, imitation of the human singing voice was one of the main aims of the original project. It employed a new synthesis algorithm developed by Xavier Rodet, FOF (*Fonction d'Onde Formantique*) synthesis or, in English, Formant Wave Function synthesis. Each FOF generator provides detailed control over the spectral shape of a single formant region. Complex timbres can be synthesised very accurately by summing the outputs of several FOF generators representing the various formant regions of the timbre required. An alternative approach, using filters to shape the spectral envelope, is also incorporated in CHANT. Although useful in many circumstances this alternative does not provide the same potential for crossing the time/frequency divide.

As well as using a new synthesis algorithm, CHANT was structured in a novel way: it used a synthesis-by-rule strategy. Many parameters were automated to respond in specific ways according to rules derived from the study of vocal behaviour. For example, a change in the amplitude of a note automatically results in changes to its spectrum. Likewise, the spectral balance of the synthesis is adjusted according to the fundamental frequency and its position in the range of the particular voice type selected (in operation such rules can be turned off).

CHANT has proved highly successful and has been used by many composers, for example, by Jean Baptiste Barrière in *Chréode I*, and by Jonathan Harvey in *Mortuos Plango*, *Vivos Voco* and *Bhakti*.

4.2. The FOF synthesis method

At first sight CHANT might appear to have very little in common with granular synthesis. Paradoxically, and interestingly for the composer wanting to explore the intersection of the time and frequency domains,

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

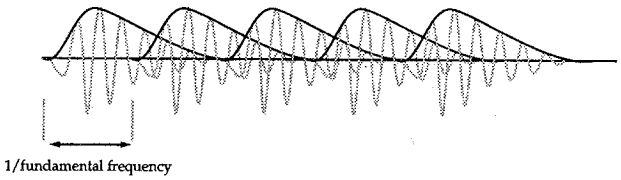


Figure 3. Successive overlapping FOF excitations.

FOF synthesis, for all the control it offers over the frequency domain, in fact works in the time domain. It generates a sequence of (usually) overlapping excitations or grains. These excitations are very short, often of the order of 0.02 s duration. Each excitation comprises an enveloped sine wave. The frequency of the sine wave determines the centre frequency of the formant region and the number of excitations per second becomes the fundamental frequency of the sound (figure 3). It is the shape and length of the envelope which determine the spectral envelope of the formant region. The envelope of the excitation essentially comprises a sigmoid rise and an exponential decay. The bandwidth (measured at -6 dB) of the formant region is determined by the rate of the exponential decay of the local (i.e. grain) envelope: the faster the decay the broader the bandwidth. The skirtwidth (at -40 dB) of the formant region is similarly related to the rise time of the local envelope: shorter rise times result in broader skirtwidths (figure 4).

Once again this demonstrates the overlap of the time and frequency domains. It may at first seem surprising that manipulating the envelope of a sound determines its spectral shape. Note, however, that the envelopes in question are extremely short (~0.02 s) and repetitive so that they might be considered an aspect of the waveform of the sound rather than an envelope in the usual sense. There is also a direct relationship with the example of the effects of filtering on envelopes given earlier. Narrower bandwidths, whether in FOFs or filters, correspond to longer envelopes.

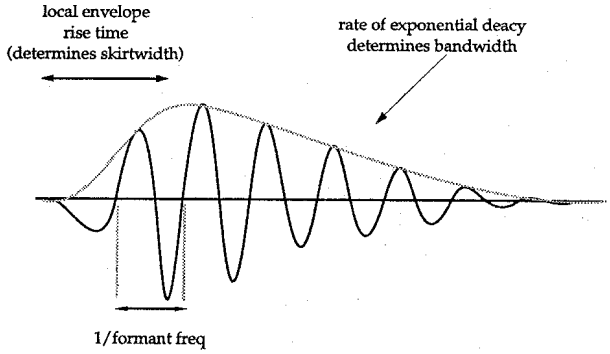


Figure 4. A single FOF excitation.

In simulating a natural timbre, such as the human singing voice, the outputs of a number of FOF generators are summed creating an overall spectrum comprising several formant regions. By controlling the centre frequency, bandwidth and skirtwidth of each formant region, detailed control over a complex spectrum can be obtained. Furthermore, the number of control parameters is manageable (unlike the situation that often arises with additive synthesis where each partial is controlled independently) and parameters relate directly to perceptual characteristics of the sound.

CHANT represented a major development in the way in which timbres could be synthesised and the richness and quality of its synthesis is remarkable. It is not simply the attention given to observing the behaviour of the human voice and other models that leads to these impressive results. It is also the combination of the time and frequency domain approaches. The frequency domain control enables the spectrum to be precisely shaped, while the time domain aspect adds variety and liveliness to the sound. Each excitation maintains the parameter values given at the time of its creation. Successive grains pick up the values of parameters at their current settings when they commence. Where a parameter is changing continuously, therefore, the result is not that all the grains change continuously, but that successive overlapping grains have slightly different (fixed) values for that parameter. In the same way that a rapid succession of subtly varying grains creates a rich, lively sound in IMPAC, so in CHANT, the sound has a complexity lacking in many synthesis methods.

4.3. FOF synthesis in Csound

My first encounter with CHANT was at EMS soon after I had begun to use IMPAC. A copy of CHANT had just arrived at EMS, and composers and researchers were investigating its potential. Working concurrently with IMPAC, and at times subverting that program to produce timbres from its granular synthesis, I perceived a clear link between the two programs. CHANT too worked with grains to produce timbres. The excitations of FOF synthesis were no more than very precisely controlled and shaped grains. This being the case, it seemed that it ought to be possible to use FOF synthesis to move between the time and frequency domains in the same way as I was attempting with IMPAC. The greater precision with which the grains were controlled would make the transformations more subtle.

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

CHANT itself, however, restricted the ways in which FOF synthesis could be used for such purposes. The very features of its synthesis-by-rule approach which made its use for timbral imitation so user-friendly limited its flexibility for granular synthesis. It was a 'black box', whose external controls could be adjusted but whose internal structure was fixed. Limited modifications could be made by means of user-subroutines, written in Fortran, but even these did not change the basic structure of the synthesis process. For example, the original version of CHANT was monophonic: all the FOF generators had to have the same fundamental frequency. In vocal synthesis this was to be expected but for granular synthesis it was a hindrance. In order to explore the full potential of the FOF algorithm as a hybrid time/frequency domain synthesizer it was necessary to extract FOF synthesis from its original surroundings and implement it in the context of a more flexible modular synthesis program. My first realisation was programmed in Macro 11 assembler as an additional unit-generator for Music 11 (Clarke, Manning, Berry and Purvis 1988). When Csound superseded Music 11 the unit-generator was reprogrammed in C (Clarke 1988) and later became an official part of Csound (Clarke 1992).

Music 11/Csound has long been renowned for its flexibility. Although it is by no means the easiest program to learn, and some consider its interface and structure outdated, it has survived and even grown in popularity because it is possible for an experienced user to create almost any sound with it. This is demonstrated by the fact that my initial prototype of FOF synthesis in Music 11 was created at the normal user level (by means of an orchestra using the existing unit-generators). Nonetheless, in order to maximise the performance of this complex algorithm, it proved necessary to create a new unit-generator specifically for FOF synthesis.

The resulting FOF module is like any other Csound unit-generator. It can be patched together with other modules in a flexible manner. The FOF unit-generator creates a single formant region, or in the terminology of granular synthesis, a single stream of grains. If these are synchronised with other FOF unit-generators, a complex spectrum can be produced with several formant regions as in the original CHANT. Alternatively, the grain streams can be independent, producing not a unified timbre but a complex granular texture. Because the difference between creating a timbre and a texture is simply a matter of the parameter settings of the FOF generators, it is possible to move smoothly between one and the other, transforming a timbre into a texture and *vice versa*. A vocal timbre, for example, might dissolve into a granular texture. Parameters such as the fundamental and formant frequencies might have their 'normal' vocal settings modified by being multiplied by time-varying functions. In perceiving a timbre the listener is primarily aware of the frequency aspect of the sound, the fundamental and its overtones. In perceiving a granular texture the emphasis shifts more towards the time domain. This shift is shown in the different terminology used by FOF synthesis and granular synthesis for the same parameters, reflecting the change in perception: fundamental frequency (FOF synthesis) becomes grain density (granular synthesis); bandwidth becomes grain envelope; formant frequency becomes grain pitch, frequency jitter becomes random grain distribution. Musically it is possible therefore to move across the perceptual divide and compose at the intersection of these domains. In other ways too it becomes possible to be more adventurous in using FOF synthesis. The flexible way in which modules can be patched in Csound means that all the main FOF parameters can be controlled by the output of other unit-generators. A limitless variety of configurations is possible ranging from timbral synthesis to complex, multi-layered granular textures. On the downside, whilst it is possible to recreate many of the automated rules of CHANT, this involves much more work than in the original program.

My interest in the theory and programming of FOF synthesis was not purely theoretical, it was driven by a creative goal. Working with CHANT and IMPAC resulted in musical ideas which required a hybrid of the two systems. These ideas were eventually realised in the piece *Mälarsång* (Clarke 1987). In this work vocal sounds transmute into other voices, perform impossible acrobatics, and dissolve into granular textures. It combines sounds created by the original CHANT program (with the addition of my own Fortran user-subroutines) and those generated by Music 11 with the addition of the FOF unit-generator. Sound example 2 is a section of *Mälarsång* in which vocal sounds transform and become submerged in a granular texture before re-emerging at the end of the example. In the music, vocal sounds frequently dissolve into, or form out of grains, either glottis clicks or bell-like sounds depending on the local envelope, and it is these grains that form the texture.

Just as it is possible to start from FOF synthesis and work towards granular synthesis, so it is possible to work the other way: to start with a granular synthesis algorithm and work towards timbral synthesis. To a certain extent this is what I attempted with IMPAC. However, most granular synthesis algorithms are not constructed in such a way as to provide sufficient timbral precision for frequency domain work. Such precision is not so significant in generating textures and brings with it the disadvantage of making the program slower to run. In FOF synthesis

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

the envelope shape is of paramount importance and must be controlled with great precision. In granular synthesis, which is time domain oriented and less concerned with the resulting spectrum, less sophisticated envelopes are frequently used and there is often less precise control over their timing.

In FOF synthesis it is also necessary to calculate the onset time of the grain precisely. In granular synthesis it is not normally necessary to work at a precision greater than the sample period, in fact some programs generate grains only at a slower control rate. While this is fine for statistically distributed grain textures it does not provide the precision necessary for timbral synthesis. The timing of grains determines the fundamental period in FOF synthesis and all aspects of the grain/excitation need to be calculated with a resolution finer than the sampling rate. If a grain is scheduled to start in the middle of a sample period this is taken into account by the algorithm, and the grain parameters are adjusted to compensate: the phase of the formant wave and of its envelope are fractionally incremented to reflect the situation. FOF's precision gives more flexibility: it can still operate successfully as a granular synthesis generator but can also slide seamlessly from this into generation of precise timbres. FOF synthesis can bring together the time and frequency domains in a way that is rarely possible in other forms of synthesis. In particular, the composer can play with the ambiguity that arises at the intersection of these two modes of perception and merge aspects of each.

5. GSAMX: TEMPORAL TRANSFORMATIONS OF FREQUENCY

Early interest in granular synthesis was based on grains synthesised with sine waves or other waveforms, or with FM generators. More recently much interest has been shown in the granulation of prerecorded sounds. In other words, the grain is not based on a synthesised waveform but on a segment of a sound file. One example of such an approach is the GSAMX program by Barry Truax.

Barry Truax has undertaken much important work in establishing granular synthesis amongst the repertoire of computer music techniques. As a composer his recent compositions have explored a variety of approaches to granular synthesis resulting in a highly original musical style. As a programmer he has pioneered new granular synthesis systems. His system at Simon Fraser University in Vancouver has evolved to incorporate many different approaches. These include additive synthesis of grains, as well as frequency modulation options, and most recently the granulation of sound files. Currently the program runs on a PDP 11 computer controlling a DMX-1000 processor.[2] Prototypes of new versions of the system have been built in the form of additional hardware controlled from Atari and Macintosh computers. Plans to make such versions available more widely are under consideration.

In the PDP 11 version, real-time control over the synthesis parameters is effected from a single command line; the user scrolls backwards and forwards across the command line on the screen and selects and adjusts parameters. Particular combinations of settings can be stored as presets and then recalled with a single keystroke. User-defined 'ramps', to transform sounds dynamically, can also be specified and called into action by the user in real-time. However, in terms of external devices (joysticks, etc.) real-time control over the system is not as far developed as at EMS. The Atari and Macintosh versions make use of the mouse to control faders on the screen to adjust the parameters. This makes user interaction more intuitive although only one parameter can be adjusted at a time in this way.

Using a programmable synthesis engine, the DMX-1000, has advantages over the custom-built oscillators at EMS. Higher densities of grains are possible and, although there is (as with IMPAC) still a division between the controlling system and the sound generation system, the synthesis engine being in software is more flexible and programmable. The higher grain densities are significant, making it possible to move much more easily from grain streams where grains are individually perceptible, to those that form 'granulated' textures, to streams that blend totally into a texture or timbre. It is possible, therefore, to move from perception focused in the time domain to that focused in the frequency domain.

The development of GSAMX, the version of the program for granularisation of sound files, adds another dimension to the possibilities of the system. Grains comprise short enveloped fragments of a sound file rather than a sine wave or FM synthesis. Part of the interest in this approach has been because samples or longer sound files can provide a rich source of material for generating granular textures. In addition, however, it is possible to use granulation of sound files as a means of transforming the original sounds, as a form of digital signal processing. Whereas the essential character of original sound material may disappear when the emphasis is on the time domain, it is usually retained in the transformation when the emphasis is on the frequency domain. The two alternatives demonstrate the extremes of the time domain approach and the frequency domain approach, but there are many intermediary positions and these are possibly the most interesting.

[2] The DMX-1000 is a programmable, high-speed digital signal processing unit controlled from a host computer.

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

As with phase vocoding, granulation can be used for a form of time stretching or pitch shifting: the speed and frequency of the original sound can be adjusted independently. Truax recommends that grains should generally be longer (40–50 ms rather than 10–30 ms) when working with pre-recorded sounds (Truax 1994). This is so that the frequency components of the original sound file can be captured in the grain. This can be compared with choosing an appropriate window length for phase vocoding. In phase vocoding it is usual to choose a window length in relation to the period of the fundamental frequency of the sound. Grain/window lengths for time stretching and pitch shifting, whether using granulation or phase vocoding, should ideally be chosen so as to be long enough to contain pitch data from the signal (i.e. no shorter than the fundamental period) but not so long as to contain significant temporal transformations. In both cases longer grain or window lengths also have the advantage of minimising the side effects of the processes.

Ideally, therefore, the temporal and timbral aspects of the original sound file are separated through granulation. The temporal evolution is represented by the movement from grain to grain, whereas the timbral aspect is captured internally within each grain, and this makes it possible to alter the speed and frequency independently. If the rate at which successive grains progress through the sound file is altered, the temporal evolution of the sound will be stretched or contracted. If the rate at which sound file data is read internally within the grain is altered, the pitch of the sound will rise or fall. Each grain has a starting point for reading data from the file. It is the rate at which these starting points progress through the file that corresponds to the speed. Each grain then reads sound from its starting point; the rate at which it reads within the grain corresponds to the pitch (figure 5). Since the waveform within each grain contains timbral rather than temporal information, Truax argues that the direction in which the data is read at this level is not significant and his program usually reads the data in reverse.

It is possible, therefore, with granulated sound files to vary pitch and speed independently, as with a phase vocoder. All such processes introduce side effects and modify the spectrum (Jones and Parks 1988), resulting in particular from phase discontinuities. Generally speaking, granulation is not as pure as phase vocoding and introduces more changes in the spectrum. However, if the process is used for compositional purposes and not simply to modify the sound, the transformations produced by granulation can be an interesting and positive aspect of the procedure. Potentially it is possible to move continuously between such uses, a form of digital signal processing, and the creation of granular textures. Once again

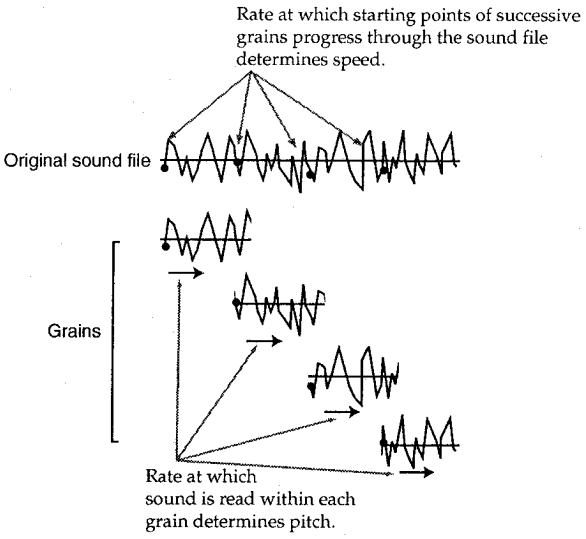


Figure 5. Granulation of a sound file.

boundaries cannot be drawn precisely: there is a sense in which all the sounds generated with such a process are signal processing, and also a sense in which all are granular synthesis. It is once more a question of perceptual focus: is the listener more aware of new textures being synthesised or of a recorded sound source being processed? This is an exact parallel to the continuum described earlier between granular (time-oriented) synthesis and spectral (frequency-oriented) synthesis. Again, traditional granular synthesis programs with their orientation towards the time domain often do not provide the precise spectral control required if one is to move convincingly across the boundary. The ability to combine granular synthesis using sound files as source, with greater spectral control, becomes possible when one merges the design of a granular program such as GSAMX with the closely related, frequency-domain-oriented (at least in terms of perception) FOF algorithm.

6. FOG: TIMBRAL CONTROL OF TIME

It is possible to granulate sound files using the FOF algorithm as a model. I first proposed this approach and implemented it on the transputer version of Csound (Bailey, Purvis, Bowler, Manning and Clarke 1990) in parallel C in 1993. More recently Gerhard Eckel and Manuel Rocha Iturbide have programmed FOG (FOF Granular) synthesis for Max/FTS on the IRCAM Signal Processing Workstation (Eckel, Iturbide and Becker 1995) and I have implemented it in C as an addition to the Unix and PowerPC versions of Csound.

Traditional granular approaches are capable of producing convincing results in terms of generating textures from sound files. However, when perception is focused in the frequency domain, the finer control

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

of the FOF algorithm is a distinct advantage. Such spectral control is not normally part of the intention or design of granular programs which are intended for granular synthesis – primarily a time domain activity. The envelopes which shape the grains in GSAMX, for example, are linear and their shape cannot be modified in detail. The only control is over the proportion of the grain length that is taken up by the linear rise and decay of the envelope. Both rise and decay are adjusted together and cannot be controlled independently. As explained above, in relation to FOF synthesis, the local envelope shape is important in determining the spectrum of the sound. The envelopes perform a kind of complex amplitude modulation on the original signal. Linear envelopes have second-order discontinuities and their spectra are rich in high frequencies. When such signals modulate another signal this high frequency content is multiplied resulting in distinctive timbral qualities. The FOF envelope is smoother and does not contain so many high frequencies. It is more flexible, permitting more control over the spectrum than most granular programs.

Another aspect of synthesis in which the FOF algorithm provides greater precision than is normally required for granular synthesis is the timing of grains. As already described, in creating textures most granular approaches do not require the precision of the timing of grains to be finer than one sample period. GSAMX, for example, is only able to create new grains at clock interrupts every 1 ms. While this is not significant in terms of statistically distributed granular textures, it is of significance in the frequency domain. In the granulation of pre-recorded sounds, precise timing becomes important when the focus of attention is timbral, frequency oriented, rather than textural, time oriented.

6.1. The FOG synthesis method

The basic FOF algorithm is little changed for FOG synthesis. Instead of using a sine wave from a stored lookup table for the waveform of each grain, a sound file (or a sound file segment) is read into the lookup table (Gen01 in Csound already permits this). The increasingly large amount of RAM available on most computers today means that storing a sound file as a lookup table in RAM does not prove too much of a constraint. (The alternative of buffering the sound file data and reading segments into RAM, although more open ended, is much more time consuming and not worthwhile.) The changes to the algorithm for FOG synthesis have more to do with how the unit-generator is controlled rather than with any change to the algorithm itself. It is instructive to compare a selection of significant input parameters for the two unit-generators:[3]

FOF: xfund (—) xform koct kband kris kdur kdec ifna ifnb
FOG: xdens xspd xfrq koct kband kris kdur kdec ifna ifnb

The fundamental/density control (**xfund/xdens**) works identically in both modules, the change in terminology simply reflecting the timbral and granular usage. In FOG, the speed parameter (**xspd**) determines the rate at which the grains move through the lookup table containing the sound file data. At normal speed the grains will move through the sound file at the speed it was originally recorded. Slower or faster speeds will time stretch/contract the file as described above. For reasons that will be discussed below this is implemented as a phase control. It is possible for the phase to decrement rather than increment and for the sound file to be read backwards. In FOF synthesis, since the waveform in the lookup table is a sine wave and new grains always read from the same position in the table, no equivalent FOF parameter exists. In FOF synthesis the formant frequency (**xform**) is the frequency at which the stored lookup table is read. This becomes the central frequency of the formant region when the sound is produced. Likewise in FOG synthesis the frequency parameter (**xfrq**) determines the speed at which the stored lookup table is read within each grain, but in this context the result is normally perceived as the frequency of the processed sound relative to that of the original sound file. It is therefore implemented as a ratio: 1 being the original speed, 2 and 0.5, for example, being an octave higher and an octave lower, respectively. Negative values cause the sound file to be read in retrograde within each grain.

Octaviation (multiple) works identically in both modules (**koct**). In FOF synthesis it produces a rather unusual method of octave transformation of the fundamental (Clarke *et al*. 1988: 360). In FOG synthesis the same process results in the sound file dissolving. The **kband/kris/kdur/kdec** parameters control the envelope of each grain (the 'local' envelope) and are identical in operation for both FOF and FOG synthesis. When FOF synthesis is used to generate timbre they determine the spectral envelope of the formant region (as described above). In FOG synthesis the perceptual meaning of the grain envelope depends on the context. It is possible for the grains to overlap and for symmetrical envelopes to cancel each other out in such a way that the original sound file is recreated. The grains are then transparent and there is then no spectral or other implication. There

[3] These are not full Csound specifications: only the relevant input parameters are shown. Following Csound convention, the first letter of each parameter (x, k or i) indicates whether the parameter will accept input data at audio or control rates or only at initialisation.

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

is an implication, however, as time stretching or pitch shifting takes place. This is because side effects are introduced by the overlapping grains containing different portions of the sound file. The shape and length of the attack and decay portions of the grain envelope have an effect on this overlap and its spectral result. If the sound file is dissolved into a series of separate grains (either by reducing the density in the normal way or by multiple octaviation), the grain envelope changes from a frequency domain feature to a time domain feature and is heard as an envelope (of amplitude) in the normal way (the longer the grains the more clearly this will be perceived).

The final significant parameters in this context, **ifna** and **ifnb**, are the identifiers for the function tables to be used respectively for the grain waveform and for the envelope rise and final decay. In FOF, **ifna** references a lookup table which is normally a sine wave, whereas in FOG it stores data from a sound file. This is the crucial difference between FOF and FOG as already discussed. The rise portion of the grain envelope (**ifnb**) is normally sigmoid (a unipolar segment of a sine wave). The same waveform is used in reverse for the final decay of the envelope, rounding the exponential decay to zero to avoid any discontinuity. Other shapes may be used (for example linear envelopes are frequently used in granular synthesis), but these will add new elements in the spectrum and lose the spectral precision which has been described above as one of the main features of using both FOF and FOG as the basis of granular synthesis.

7. TIM(br)E: COMPOSING WITH AMBIGUITY

7.1. New control strategies

It has already been demonstrated that there is a significant overlap between the time and frequency domains: the distinction is a matter of perspective rather than absolute. One of the situations in which the distinction becomes apparent in FOF and FOG synthesis is in the way in which formant regions and grain streams are controlled. (Formant regions and grain streams are in fact the same thing viewed from the different perspectives of frequency and time, respectively.) From the frequency domain perspective, control over the grains is in terms of fundamental and formant frequencies, bandwidth, etc. These controls can be implemented as straightforward input values: constants or variables in Hertz (as is done in the Csound implementation of FOF, for example). From the hybrid frequency/time perspective there are advantages in implementing the fundamental and formant frequencies in another way, by means of a phase control. The reason for this alternative strategy is that working at the intersection of the two domains means that it is useful to be able to control grain streams (i.e. FOF or FOG unit-generators) independently and yet have the option of bringing them back into phase at certain moments. Earlier (section 4.3), an example was given of how a single voice simulation in FOF synthesis might dissolve with different formant regions having their own fundamental and formant frequencies. This, it was suggested, might be achieved by multiplying the parameter values used to create a coherent vocal simulation by different time-varying functions. Suppose all these time-varying functions eventually converge on the value 1, in other words the original values are restored. With a simple vocal imitation the original timbre will return. Although the fundamental phase will no longer be synchronised, this is not likely to have a significant effect in most circumstances (the formant phase in FOF synthesis is reset at each fundamental period and so is not a problem) [sound example 3].

If a similar process of divergence and convergence is applied to a FOG synthesis, where the time domain is of greater significance, the issue of phase becomes critical. For example, one might play two identical FOG grain streams, derived from a sound file, out of the two channels of a stereo output. Initially the identical streams might both have settings such that they reproduce the original sound file unaltered. The two streams might then diverge, each transforming the sound file in different ways, changing the density, pitch and speed independently. If one were now to try and bring the two streams back into line, any phase difference would be of far greater significance. In particular, the speed parameter determines the phase of the function table (i.e. the position within the original sound file) at the start of each grain. Even if the speed in the two streams is identical, there is no guarantee that the phase will be synchronised if a simple speed ratio control is used. Only by adapting this control to one that inputs phase directly can resynchronisation be guaranteed. With a phase control input a single 'phasor' unit-generator can be used to produce the phase for both streams. At the point where the two streams diverge, each FOG generator can take this phase and modify it independently with a time-varying function. Thus, the speeds and phases of the two streams will diverge. However, as soon as the time-varying functions converge, not only do the speeds of the streams return to the same value, but phase synchronisation is also guaranteed [sound example 4].

In certain situations it is also important to be able to resynchronise the fundamental phases of FOF or FOG streams. For example, two streams of grains are at a density where the individual grains can be heard distinctly. At first the grains occur at regular time intervals and the two streams are synchronised. Sup-

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

posing one wishes to make the two streams diverge in their density, one perhaps increasing and decreasing its density in a periodic fashion controlled by an LFO, the other changing at random. Later one wishes to remove the LFO and the random element from the two streams and resynchronise them. With a simple frequency control over the density, although one might make the density/fundamental frequency the same again, the phase of the two streams would be different. Again, to achieve this, control needs to come from a phasor and, because of the particular nature of the fundamental/density parameter, a special unit-generator is also needed to produce a control message to trigger new grains. This approach has been used in some experimental versions of the Csound FOG unit-generator in place of the more normal frequency control.

7.2. Composing at the intersection

The above examples, although simple in themselves, demonstrate ways in which apparently unrelated streams of grains can be synchronised. Extending this principle to a larger scale, it is therefore possible to compose with multi-layered granular textures from which, by synchronising some or all of the grain streams, clearly defined timbres can emerge. The basic material of the composition can itself dissolve and reform. A composition can be considered as a sea of grains which continually form and reform into a variety of configurations. All the material is unified by its common source in this sea. Musical objects, rather than having rigid definitions, are fluid, momentary configurations of elements that will dissolve and reform into other objects. The distinction between foreground and background, between timbre and texture, is also fluid and subject to continual changes of perspective. Of course, ambiguity and fluidity have always been characteristic features of music. Its ability to transcend rigid distinctions is one of music's most distinctive and important aspects. In much twentieth century music, motivic and thematic material has been particularly malleable. Working at the intersection of time and frequency offers new dimensions of metamorphosis. The basic material itself, notes and timbres, becomes ambiguous. A single note may transform seamlessly into a vast texture, or a chord emerge from within a complex web of grains.

Such methods lead to new approaches to composition. Traditionally, composition has been an additive process, at least in construction if not in conception, the result of putting notes together to form larger musical objects (chords, motivic material, etc.), which themselves are added to form larger structures. Working with granular synthesis (e.g. IMPAC or GSAMX) can lead to a different, subtractive approach. Musical form is created by sculpting a shapeless, random mass of sound. Shape is the result of imposing boundaries on the sound, in effect cutting away material, carving a form out of the material. This approach was particularly striking when using IMPAC because initially the program generated random events without restriction until the composer began to shape the sound. Composing at the intersection of time and frequency perhaps implies other metaphors for the compositional process. The pull of gravity drawing material together into different, varying formations, or a stream where patterns, whirlpools and currents form out of a constantly flowing mass of water.

Composing in an environment where the basic materials are fluid puts greater emphasis on the process of transformation and less on the statement of musical objects. Relativity takes the place of fixed definitions and hierarchies. How such music is perceived is, of course, a vital question for the composer. What is a listener capable of perceiving and how can a music of continual transformation be shaped meaningfully? Perhaps part of the answer is that the degree of transformation and change can itself become a parameter of the musical structure, sections of relative definition contrasting with more fluid passages.

To be able to develop and work with such processes it is essential to have a reasonably fast cycle of aural feedback. The composer must be able to imagine a transformation, program it and hear the result within an acceptable time span. Until recently the complexity of the FOF/FOG algorithm made this impossible. Previously, many experiments I tried took hours or even days to calculate. However, it is now possible to generate many of the simpler examples in real-time using a desktop computer. Even more complex transformations are usually calculated quite quickly. For example, to give some idea of the scale of developments, a thirty second orchestra and score that ten years ago took about fifty-four hours to calculate on a PDP 11 computer, will now (1996) run in less than one minute on a Macintosh 120 MHz PowerPC. It is now possible to investigate these techniques more easily, and such developments form the basis of my next composition.

Throughout this project technology and creativity have interacted, developments in one area inspiring new ideas in the other. It is important that electroacoustic music is creative with technology and does not simply use it unthinkingly. To divide the music from the technology and to restrict creativity simply to the 'musical' aspect is a mistake. Computer music should be seen as a whole, creativity permeating all aspects of it. New technology will therefore inspire new musical developments and *vice versa*. Technology has reached the point where it is possible to combine many of the best features of programs such as

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press

IMPAC, GSAMX and CHANT, and to work on a home computer at speed. The potential for exploring sound at the intersection of the time and frequency domains is increasing and with that come new possibilities for composing and listening.

SOUND EXAMPLES

Sound example 1. An extract from *Uppvaknande* by Michael Clarke (2′13″ to 4′15″) demonstrating the use of the program IMPAC to produce timbres as well as granular textures.

Sound example 2. An extract from *Mälarsång* by Michael Clarke (2′29″ to 6′09″). This extract demonstrates the use of FOF synthesis to produce both granular textures and timbral synthesis (especially vocal imitation). It illustrates the possibility of moving continuously between timbral and textural synthesis, freely crossing the divide of the time and frequency domains.

Sound example 3. A synthesised vocal timbre (FOF) dissolves into a granular texture with several independent streams before reforming as a voice.

Sound example 4. A bass clarinet recording is granulated (two versions) using FOG. The granulation divides into five streams which diverge and reform. In the first version a frequency control is used for the 'speed' parameter and phase synchronisation cannot be restored. In the second version phase control permits exact resynchronisation. (Bass clarinet: Esther Collard)

REFERENCES

Bailey, N., Purvis, A., Bowler, I., Manning, P., and Clarke, J. M. 1990. Concurrent Csound: Parallel execution for high speed direct synthesis. *Proc. Int. Computer Music Conf.*, pp. 46–9. Glasgow, Scotland.

Clarke, J. M. 1984. *Uppvaknande*. CD recording: MPSCD003 (1995). Impetus Distribution, London.

Clarke, J. M. 1987. *Mälarsång*. CD recording: MPSCD003 (1995). Impetus Distribution, London.

Clarke, J. M. 1988. *FOF synthesis on the Atari ST*. Composers' Desktop Project, Heslington, York, England.

Clarke, J. M. 1992. An FOF-synthesis tutorial. *Csound manual*, pp. 107–10. Cambridge, MA: MIT Media Lab.

Clarke, J. M., Manning, P. D., Berry, R., and Purvis, A. 1988. VOCEL: New implementations of the FOF synthesis method. *Proc. Int. Computer Music Conf.*, pp. 357–71. Cologne, Germany.

Eckel, G., Iturbide, M. R., and Becker, B. 1995. The development of GiST, a granular synthesis toolkit based on an extension of the FOF generator. *Proc. Int. Computer Music Conf.*, pp. 296–302. Banff, Canada.

Gabor, D. 1947. Acoustical quanta and the theory of hearing. *Nature* **159**(4044): 591–4.

Jones, D. and Parks, T. 1988. Generation and combination of grains for music synthesis. *Computer Music Journal* **12**(2): 27–34.

Roads, C. 1978. Automated granular synthesis of sound. *Computer Music Journal* **2**(2): 61–2. Reprinted in Roads, C. and Strawn, J. (eds.). 1985. *Foundations of Computer Music*. Cambridge, MA: MIT Press.

Roads, C. 1988. Introduction to granular synthesis. *Computer Music Journal* **12**(2): 11–3.

Roads, C. 1991. Asynchronous granular synthesis. In G. De Poli, A. Piccialli, and C. Roads (eds.) *Representations of Musical Signals*. Cambridge, MA: MIT Press.

Rodet, X., Potard, Y., and Barrière, J. B. 1984. The CHANT project: From the synthesis of the singing voice to synthesis in general. *Computer Music Journal* **8**(3): 15–31.

Truax, B. 1988. Real-time granular synthesis with a digital signal processor. *Computer Music Journal* **12**(2): 14–26.

Truax, B. 1994. Discovering inner complexity: Time-shifting and transposition with a real-time granulation technique. *Computer Music Journal* **18**(2): 38–48.

Xenakis, I. 1971. *Formalized Music*. Bloomington, IN: Indiana University Press.

https://doi.org/10.1017/S1355771896000052 Published online by Cambridge University Press