Xavier Rodet, Yves Potard, Jean-Baptiste Barrière
IRCAM
31, rue Saint-Merri
F-75004 Paris, France

The CHANT Project: From the Synthesis of the Singing Voice to Synthesis in General

Introduction

The CHANT project was originally concerned with the analysis and synthesis of the singing voice. This work led to a complex program of voice *synthesis-by-rule*: CHANT. This program was enriched with a constantly expanding software environment, consisting of both analysis and composition programs. In time, broader aims than the synthesis of the singing voice imposed themselves. These aims centered on the search for models of the processes involved in the production of musical sound. Our present research encompasses the physical description of sound phenomena (the *sonic material*), the articulation of these phenomena (*organization*), and compositional issues.

This research is intended to transcend simulation. Our goal is to extrapolate new creative models for music on the basis of *knowledge models* developed using the synthesis-by-rule methodology. In this research, synthesis is the proof of both our understanding of sound phenomena, and of the music itself.

In this article we reexamine music synthesis in the following way. In the first part we reconsider the development of past synthesis techniques and programs. We explain our reasons for starting from a physical model of sound production—the voice—because of its generality and its complexity.

In the second part we present and compare two types of synthesis implementations inspired by the vocal model: one based on filters and another based on *formant wave functions*. (Those who are not concerned with the details of implementation problems, as well as readers without a scientific background, can skip this part.)

In the third part we describe the CHANT synthesis-by-rule program and methods for controlling this technique. Finally, we give examples of works realized with CHANT in the context of the rule-based FORMES system (Rodet and Cointe 1984).

From a Reconsideration of Synthesis Techniques and Programs to the Choice of a Production Model

A Reconsideration of Synthesis Techniques and Programs

Let us examine some of the reasons behind the development of the established digital sound synthesis techniques and programs.

The Imitation of Analog Techniques

Paradoxically, analog devices have often been simpler to use than many programs. Analog modules can be linked to each other by a simple cable. (This analogy does not work for subprograms because of complications with argument passing.) In systems that imitate analog synthesizers, effects are almost always obtained directly, without resorting to mysterious code, and control is achieved by such simple means as turning a knob. However, this approach makes meager use of the immense possibilities for controlling digital signals, and ignores the symbol processing capabilities of computer languages.

Speed of Calculation

This concern was particularly justified at the beginning of computer music when hardware was slow and not specialized for synthesis. But an emphasis

on speed of calculation can have several negative consequences.

First, a technique that aims at rapidity is generally not related to the properties of perception. Such techniques may result in great difficulties in controlling the perceptual characteristics of sound, since they depend on the parameters of the synthesis method through a very complex and arbitrary set of relationships. Learning a synthesis method of this type is difficult and unjustified.

Second, a technique aimed at speed usually bears no relation to the mode of production of natural sounds. (There is no physical model.) This also makes it more difficult to use, because we cannot then use our knowledge of the relations between the variations of the mode of production and the corresponding variations of sound. Moreover, the development of electronic components now permits considerable computing power for a relatively low cost. The limit of possible calculations is therefore appreciably extended.

The "Patch/Note List/Function" Representation

This representation has several consequences. One of the difficulties lies in making explicit the intermediate levels of control between the patch and the note-list. A patch can be a fairly satisfactory description of an instrument or a synthesizer. But on the other hand, the patch languages that exist are weak in their ability to specify the elaborate control levels that resemble interpretation by an instrumentalist, for example, expressiveness, context-dependent decisions, timbre quality, intonation, stress, and nuances.

In these languages, to offer the real possibility of playing with sound, it is necessary to construct a very complex instrument controlled by a considerable number of parameters—to the point that writing a single note becomes an awesome task. Further, when the computation is performed on a sample-to-sample basis, the cost of calculation time must be added to the unwieldiness of the description.

Control functions have relatively slow variations, around 100 Hz on average. In the languages of the Music V family this type of control is also effected by functions. But the latter are then no longer algorithmic descriptions, as is the patch. Therefore, correlations between parameters can no longer be taken into account. Moreover, their temporal scope is subject to the length of the note. This obliges a kind of acrobatic manipulation for the continuous control of parameters by means of notes (for example in the case of a legato) or the use of functions where the desired scope differs from the length of a note (for example, a crescendo over a whole phrase). In these languages, there seems to be a lack of intermediate control levels between the patch and the organizational level of the note. We want to be able to describe these intermediate levels in a programming language, enabling interactions between sound and perception, and between sonic material and musical organization.

These remarks imply that we have to make use of computer instruments that are at least as immediate in response as analog hardware—and even more so, since the software can provide an almost limitless "intelligence." We also have to try to take advantage of the specific richness of the various techniques by choosing to exploit their idiomatic qualities. At the same time, it is essential that users have a consistent formal system at their disposal to integrate different techniques, so as to eliminate the difficulties of access and understanding specific to each method.

Finally, we must be able to implement an interaction between interpretation and microsound organization. There should not be fixed sounds on the one hand, and an external structure organizing them on the other. On the contrary, the sounds must be formed as a function of their place in a certain context. Thus, sounds are simultaneously the objects and subjects of the organization.

A Voice Model, A Synthesis-by-Rule Methodology

These are the considerations that have led us to our choice of a production model: the voice, and our choice of a methodology: synthesis-by-rule. The choice of the voice as a model of production was imperative because of its extreme richness. By the wealth of its output and the variety of musical

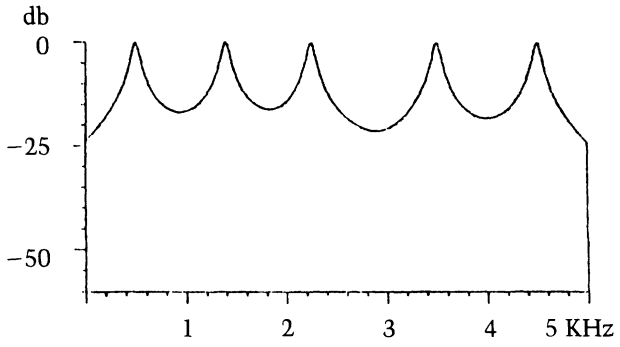


Fig. 1. Log magnitude spectrum of the transfer function H1.

and linguistic uses to which it gives rise, the voice inspires a more general and fertile approach than the study of any other instrument—no matter how complex. It is the need to account fully for the complexity of the variations in the vocal model, and more particularly, in the resonator, which obliges us to reach a level of generality that has also enabled us to move toward quite different models.

A synthesis-by-rule methodology implements complex control levels, and constitutes a formalization of musical production and composition in terms of models that can be built up incrementally.

Description of the Working of the Vocal Apparatus: A Production Model

The sonic wave of the voice is produced by a stream of air breathed from the lungs into the vocal tract, through the larynx to the lips and nostrils. At various points, the wave is disturbed by sonic sources.

One source of disturbance is the vibration of the vocal cords, modulating the stream of air breathed out through the larynx. The sounds produced are quasi-periodic and are said to be *voiced*. A second source comes from a narrowing of the buccal cavity at certain points: the lips, tongue palate, and the glottis for whispered sounds. The stream of air becomes turbulent and produces an aperiodic sound known as a *fricative*. A third type of source is obtained through the interruption of the air stream by closing the buccal cavity with the lips or the tongue, and suddenly releasing them. The noise of an explosion is thus produced and the sound is known as a *plosive*.

The sounds that come from these sources are modified by the vocal tract itself which acts as a set of resonators by filtering certain frequencies to a greater or lesser extent; this is known as the *transfer function* of the vocal tract. During the production of a sentence the vocal tract is continually changing its shape and therefore its transfer function, consequently this function can only be defined at a given instant. Moreover, each type of source can act more or less independently, for a duration and with characteristics that also vary continuously. A spoken sentence is the end product of this complex set of actions. The corresponding sound signal comes from successive phenomena, vowels and consonants that interact (coarticulate) in such a way that one cannot set a precise boundary between them (Rodet 1977; Sundberg 1978, 1979).

The production of speech, like that of numerous sounds or signals, is often represented by a model composed of a non-coupled source of excitation and a linear filter. However, it is desirable, if not essential that couplings be taken into consideration (Carré 1981; Weinreich 1977).

The filter $F1$, usually linear, is characterized by its transfer function $H1$, which varies continuously (Fig. 1). It takes into account the characteristics of the physical system that is perturbed or *excited* (such as the vocal tract), and its sonic radiation.

In the case of the voice, the model includes a periodic source P for the voiced sounds and a random source S for the fricatives or the plosives and for the breath. A source is characterized essentially by its amplitude spectrum (Fig. 2). It can also be described by means of a filter $F2$ that represents its envelope and by a source $P2$ (and $S2$) with a flat spectrum $X(f)$. The two filters $F2$ and $F1$ in series can then be connected in a single filter F. The spectrum of the resultant wave is then the product of the spectrum of the source by the gain of the filter: $Y(f) = X(f) \cdot H(f)$ (Fig. 3).

A glottal source is a good demonstration of a model of production (Fant 1970, 1973; Rothenberg 1981). But if we limit ourselves to a model of this type, we will only obtain a limited family of sources. For musical applications, the imagination must not

Fig. 2. Log magnitude spectrum of the voiced source P.

Fig. 3. Log magnitude spectrum of the output waveform.

Fig. 4. Representation of the H *parallel filter.*

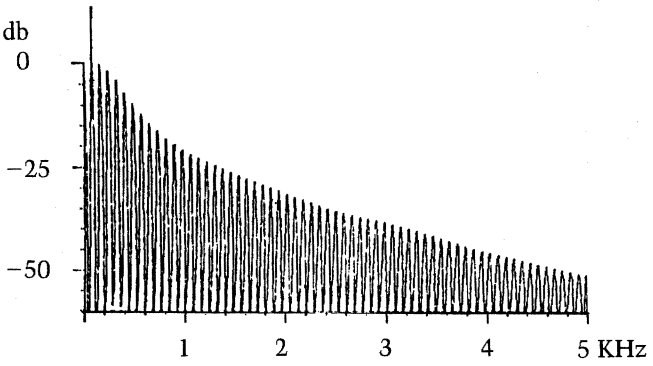
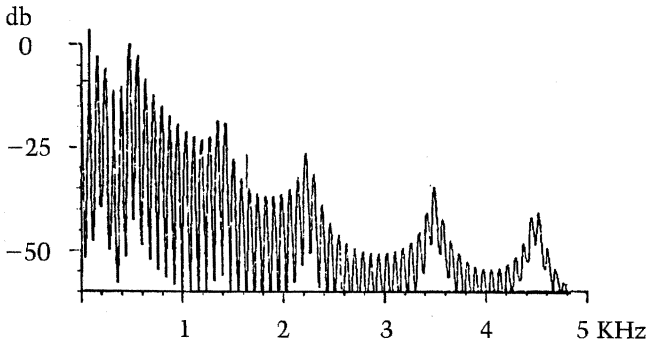


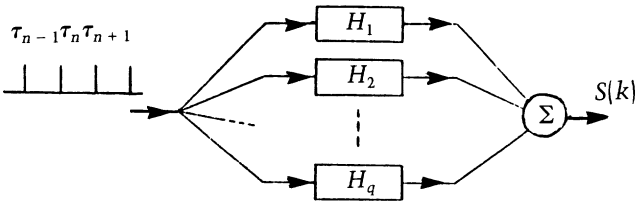
Fig. 3



be limited. It is therefore interesting to formalize this model in the spectral domain, then to include it in a more general model describing the spectrum of the source. This general model is controlled by spectral parameters, whose perceptual consequences can be easily forecast.

The vocal tract itself is a sort of tube about 17 cm long, of varying sections, branching towards the nasal tract. In a simplified fashion, it can be assimilated to a series of N resonators in series and M in parallel. Its gain or transfer function therefore usually presents N maxima, known as *formants*. The sharpness of these maxima is measured by their bandwidth at mid-height from the peak (−3 db).

The amplitude spectrum of the produced sonic wave presents maxima corresponding to the formants. The importance of these formants relates not only to the fact that they derive from the shape of the vocal tract at each moment (for each articulation) but also to their importance at the perceptual level. Because of the properties of the ear (in particular masking), the parts of the spectrum that present



maxima of this type are well-represented in terms of a formant's center frequency, relative amplitude, and bandwidth.

Next we examine the details of two implementations of this synthesis technique, by digital filters and by formant wave functions.

The Chant Synthesizer

Digital Filter

A filter such as F can be represented by its z-transfer function:

$$H(z) = \frac{\beta_0 + \beta_1 z^{-1} + \ldots \beta_q z^{-q}}{1 + \alpha_1 z^{-1} + \ldots \alpha_p z^{-p}}$$

that includes p poles and q zeros. This is the case, for example, in linear prediction (Moorer 1977). The implementation of a digital filter of this type and the calculation of its parameters α and β present difficulties that we discuss later. Consequently, we can use another form:

$$H(z) = \sum_{i=1}^{I} c_i \frac{1 + d_i z^{-1}}{1 + a_i z^{-1} + b_i z^{-2}}.$$

The H filter is then presented as a set of parallel I cells (Fig. 4). Each cell is composed of a first-order filter (a zero) and of a section of the second order (two poles) in series, with a gain c_i. Each cell is implemented quite simply. The calculation of the parameters is also much less complex, and we can directly control the perceptually interesting characteristics of the envelope of the spectrum (Fig. 5). The parameters include: a and b, which determine the center frequency of a band Δf of the envelope of the spectrum and its local form (maximum and

Fig. 5. Log magnitude of the different transfer functions of a cell.

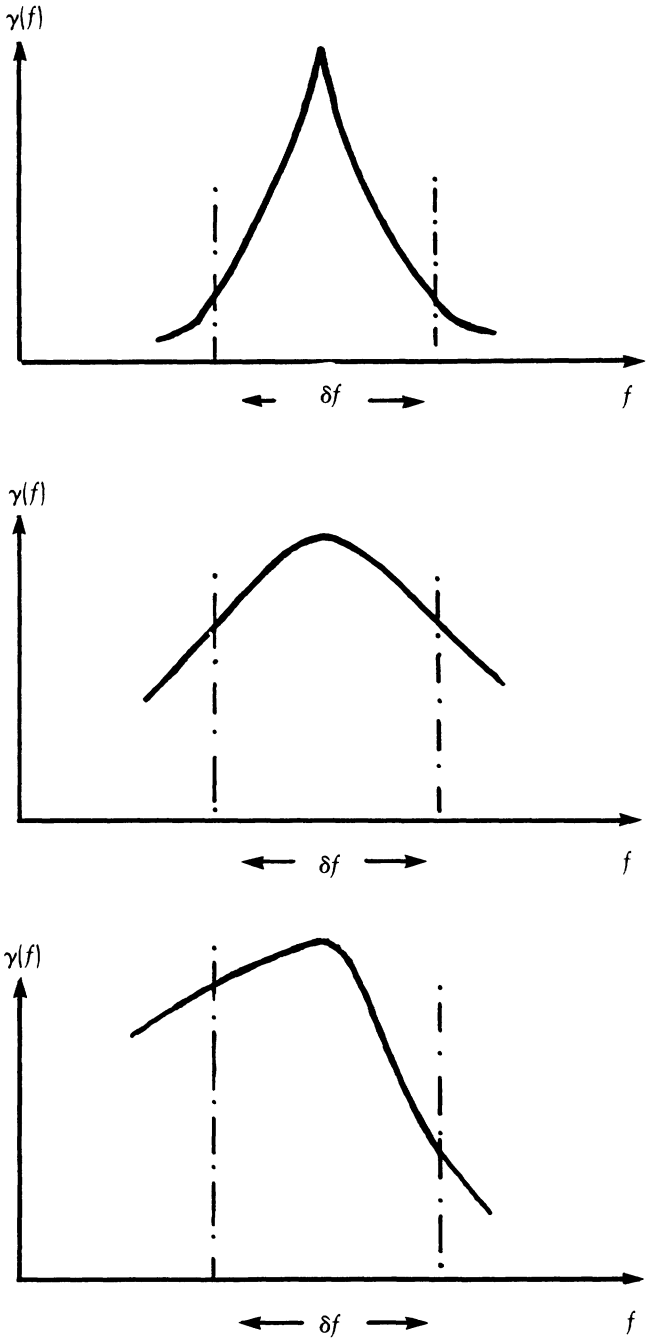
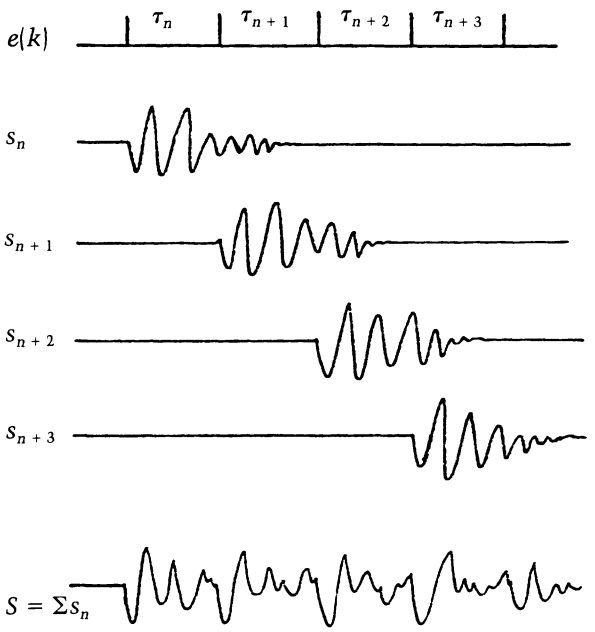


Fig. 6. Construction of the response S *as a sum of the responses* S_n.



bandwidth), *c*, which controls the global amplitude of this zone, and *d*, which enables us to change the slope of the envelope.

Formant Wave Functions (FOF)

If the excitation is a series of impulses:

$$E(k) = \sum_{-\infty}^{+\infty} e_n(k)$$

when *n* indexes the impulses in turn, then the response *S* from the previous filter can be easily calculated as the sum of the responses $s_n(k)$ shifted from a period of the fundamental $T = 1/F0$. *F0* is the fundamental frequency of the excitation and the response (Fig. 6). A response $s_n(k)$ is itself the sum of the *J* responses to the *n* parallel cells:

$$s(k) = \sum_{i=1}^{J} s_{n,i}(k)$$

where the $s_{n,i}(k)$ are termed *formant wave functions* (in French, *Forme d'Onde Formantique* or *FOF*) because they usually correspond to the formants or main modes of resonance of the system.

Fig. 7. A formant wave function (FOF).

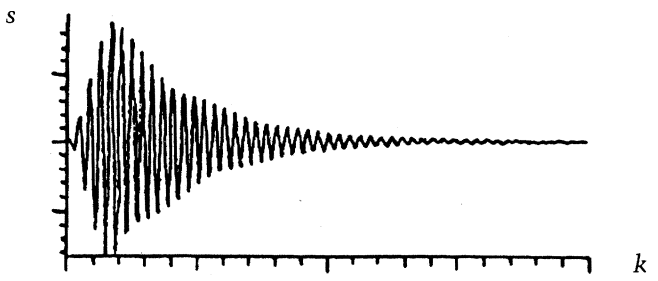
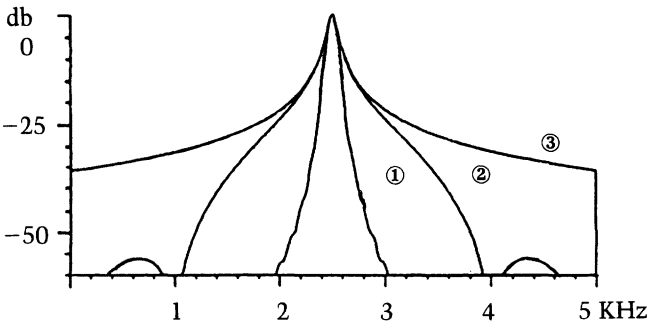


Fig. 8. Log magnitude spectrum of a FOF: $A(k)\sin([\omega_c * k]) + \Phi$ for different values of π/β. ω_c = 2500 Hz, and α/π = 80 Hz. Line 1: π/β = 10 msec; line 2: π/β = 1 msec; line 3: π/β = 0.01 msec.



Variations of fundamental frequency are obtained by changing the durations of the fundamental periods $T = 1/F0$, that is, the beginnings of the successive FOFs. Variations of the envelope of the spectrum are obtained by changing the characteristics of each FOF.

Calculation of the FOF

The response to a unitary impulse, of a cell

$$C_i \frac{1 + d_i z^{-1}}{1 + a_i z^{-1} + b_i z^{-2}} = C_i \frac{1 + d_i z^{-1}}{(1 - r_i z^{-1})(1 - r_i \cdot z^{-1})}$$

is the FOF $s_i(k) = G \cdot e^{-\alpha k} \sin(\omega k + \Phi)$, with

$$\alpha = -\frac{1}{2} \log b_i;$$

$$\omega = \text{Arg } r_i;$$

$$\Phi = \text{arc}\left[\frac{\sin(\omega \cdot e^{-\alpha})}{d_i - a_i - \cos(\omega \cdot e^{-\alpha})}\right]; \text{ and}$$

$$G = \frac{c_i}{\sin(\Phi)}.$$

Thus a FOF is obtained simply as the product of a sinusoid by an exponential envelope. However, for natural sounds, the excitation is not a unitary impulse. In order to obtain a more precise control over the spectrum, we used the following FOF (Fig. 7):

$$s(k) = 0 \quad \text{for } k < 0$$

$$s(k) = \frac{1}{2}(1 - \cos[\beta k] \cdot e^{-\alpha k} \sin[\omega k + \Phi])$$
$$\text{for } 0 \leq k \leq \pi/\beta$$

$$s(k) = e^{-\alpha k} \sin(\omega k + \Phi) \quad \text{for } \pi/\beta < k.$$

This is again a sinusoid multiplied by an envelope $A(k)$. This envelope is a damped exponential whose initial discontinuity is smoothed by multiplication by $1/2\,(1 - \cos(\beta k))$ for a duration of π/β samples. One obtains thus an envelope of amplitude $A(k)$. This envelope has an attack of a duration of π/β samples and a general damping in $\exp(-\alpha k)$. The envelope has no first or second-order discontinuity, and it can be generated very simply by table lookup or by a multiplication by $C = \exp(-\alpha)$ for $\exp(-\alpha k)$.

The amplitude spectrum of this FOF (Fig. 8) presents a maximum and can be easily adjusted with the aid of the following parameters:

- ω is the central frequency of the maximum
- $\alpha\pi$ is the bandwidth at −3 db
- β governs the skirt width or the slope of the attack

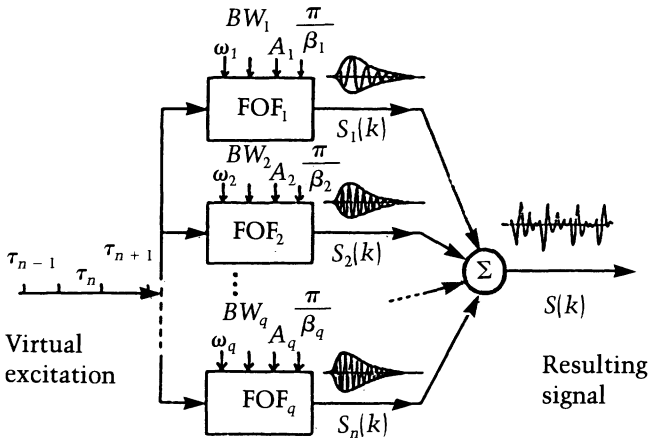
Finally, it is not difficult to adjust the amplitude of the signal produced, by means of the gain G (or, to avoid a multiplication, by the initial value of the exponential). The role of the initial phase is discussed later.

FOF Synthesizer

The structure of a FOF synthesizer is represented in Fig. 9. Its command parameters for each fundamental period are the following:

Center frequency = $2\pi\omega_i$
Bandwidth = BW_i
Amplitude = A_i
Skirt width = π/β_i
Initial phase = Φ_i

Fig. 9. Structure of a FOF synthesizer.



The FOF method presents a number of advantages as far as the filters are concerned. The precision required in the calculation is at most that required at the output (for example, 16 bits). No risk of numerical overflow exists, and the cost of the calculations is fairly low except for very high fundamental frequencies. Finally, the calculation of the control parameters is simple.

We have used this method for the synthesis of a very great number of vocal, instrumental, and other sounds. The findings indicate that one can obtain synthesis of very high quality for a low calculation cost. It is possible, for example, to perform this synthesis technique in real-time on a signal-processing microprocessor.

Comparison of the FOF and the Filter Models

The parameters of the FOF synthesizer enable us to generate the same spectrum envelopes as an equivalent filter in parallel. Thanks to the parameter β governing the skirt widths, we can even generate forms that would require a filter of a higher order, or unique functions of excitation for each cell. (See later discussion entitled "Discussion of the Parallel Synthesis Model.")

Moreover, there is an interesting difference. We have always used within each FOF $s_{n,i}(k)$ a constant frequency ω (a variable frequency can produce interesting "spectrum enlargement" effects). Since a continuously variable resonator like the vocal tract ought to be modeled by a continuously variable filter, therefore we have a ω within each FOF.

In practice, for vocal sounds and most instrumental sounds, the duration of each period is so short in relation to the speed of variation of ω that the difference is not perceptible. But in a filter the coefficients cannot be subject to discontinuity, and they have to be interpolated between each new value (whereas in FOF we need only a new value at each period, the FOF being continuous by construction).

The difference between the two methods appears when the bandwidth BW is very small and ω varies rapidly. The response of a cell in the filter is then almost a sinusoid whose center frequency evolves continuously. On the other hand, the $s_n(k)$ FOFs, triggered off at each period n, are almost sinusoids (with very weak damping if the bandwidth is very small) with frequencies that are each fixed but different from the preceding one. The spectrum then presents very dense partials giving a much richer sound than the quasi-sinusoid which comes out of the filter (Fig. 10).

This is an easy way of synthesizing a rich sound in a certain frequency band without having to specify each partial. We have, for example, synthesized cymbals in this way.

Discussion of the Parallel Synthesis Model

The parallel synthesis model (FOF and filters) previously described revealed itself as incomparably richer than might have been expected by listening to classical parallel-formant synthesizers. We discuss here some of its possibilities and certain implementation difficulties.

When the J cells are placed in parallel, one expects that the total complex gain will be the sum of that of the individual cells (or FOFs). This is not necessarily the case (Fig. 11), because some components may be opposites in phase and therefore cancel each other out. This mainly occurs between the center frequencies of two neighboring cells (Fig. 12), where one sees a "hole" (a zero) appearing in the spectrum.

Fig. 10. Comparison of the FOF and filter model outputs for small bandwidths and fast frequency variations. (a) FOF model. (b) Filter model.

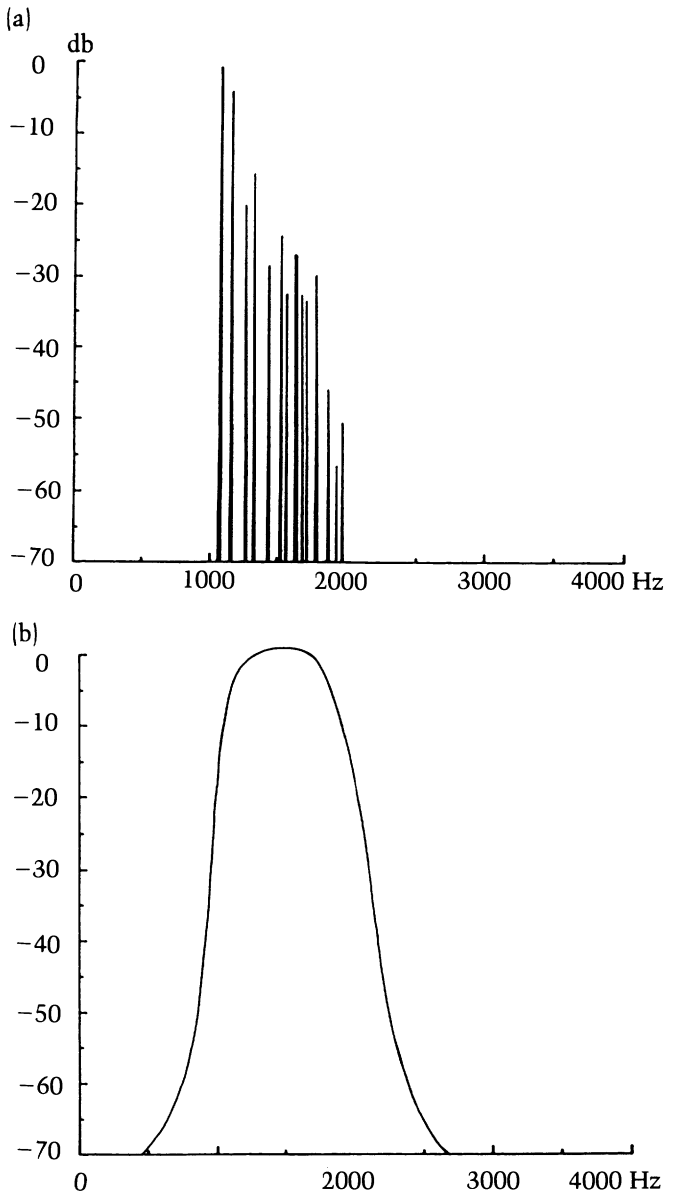
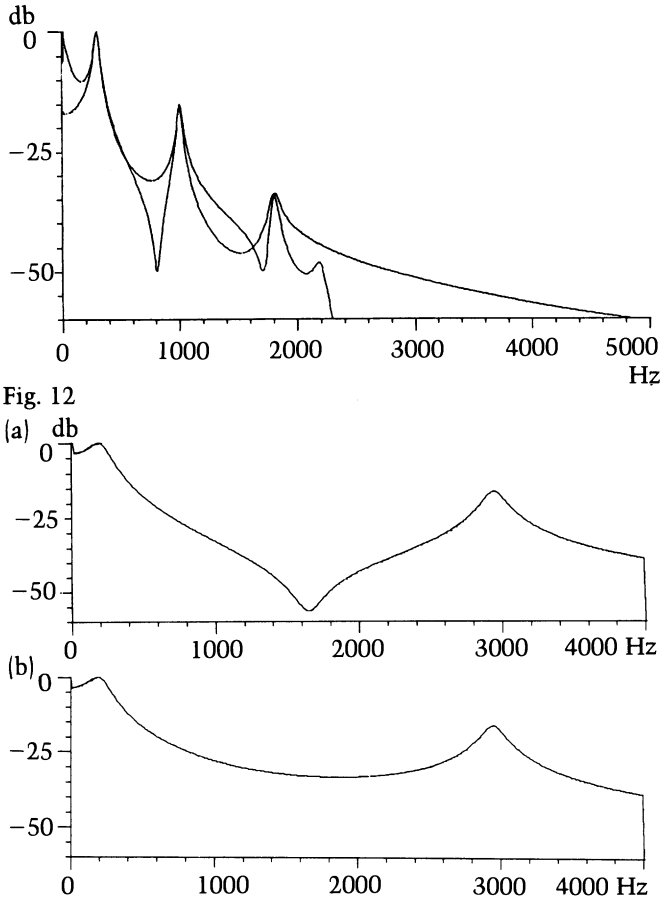


Fig. 11. Comparison of the total complex gain with the sum of the individual FOFs. Note the differences caused by phase cancellations.

Fig. 12. (a) Presence of a "zero" (attentuation) between the center frequencies of two neighboring cells. (b) Corrected spectrum.



This problem can be cured by *dephasing* the response of one cell in relation to the other (Fig. 13). This is particularly easy in the FOF method since it requires that one simply position the initial phase ϕ of the sinusoid correctly in each cell.

For the filter model, we studied a great number of configurations of poles and zeros to dephase the cells in relation to one another. But we were obliged to discard the use of dephasing filters because they were either inadequate or too complex (which would be contrary to the simplicity that motivates the parallel model). On the other hand, we can use as many sources as there are cells and their dephasing is then simple (for example, it is the initial value of the index if one uses table lookup).

This is the solution that we chose for the implementation of CHANT on the Systems Concepts digital synthesizer (also known as the Samson Box) at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University (Fig. 14). With an initial controllable phase, the generators can produce a flat spectrum and a limited periodic impulse.

This latter characteristic is also important because a parallel filter tends to have a higher gain than the series filter with the same characteristics in the frequency zone situated beyond the maxi-

Fig. 13. "Dephasing" the response of one cell in relation to another.

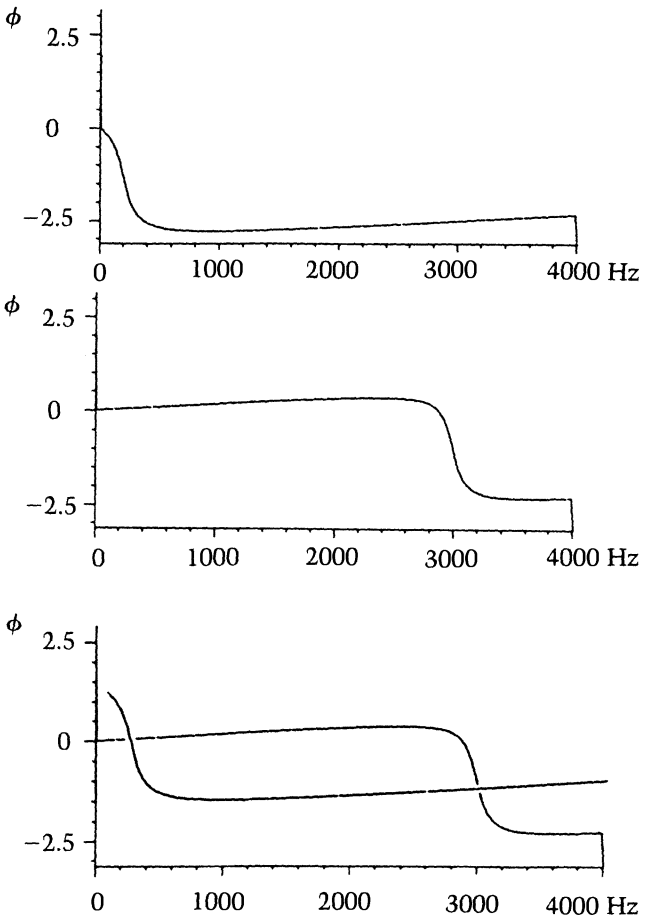


Fig. 14. Individual sources $e_i(t)$ for each cell of the parallel model.

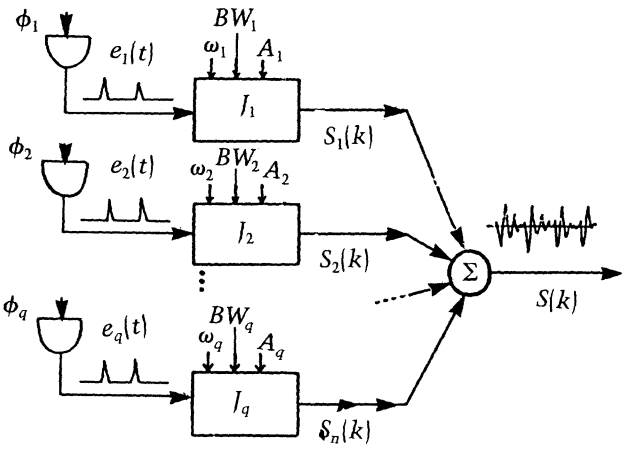
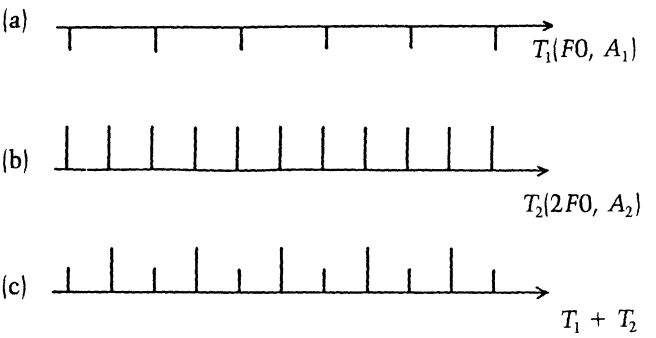


Fig. 15. A set of impulses to vary the sizes of the odd and even harmonics. (a) Odd harmonics. (b) Even harmonics. (c) Sum of odd and even harmonics.

Fig. 15



mum frequency formant (Fig. 11). In effect, the gains of the cells are added, instead of being multiplied as in a series model. The problem does not arise in FOF thanks to the control of skirt widths by β, which controls the amplitude of the spectrum beyond the maximum frequency formant.

Granting an individual source to each cell is still desirable in order to control, in each frequency band, other characteristics of the spectrum such as inharmonicity and skirt width.

One may also vary the respective sizes of a particular class of partials in relation to another: for example the even and odd harmonics. Let us suppose that a source is composed of two sets of impulses T_1 and T_2 with respective frequencies $F0$ and $2F0$ and amplitudes A_1 and A_2 (Fig. 15). The spectrum corresponding to the first is a spectrum of $F0$, $2F0$, $3F0$ partials and to the second $2F0$, $4F0$, $6F0$ partials. The spectrum corresponding to the set $T_1 \cup T_2$ is the sum of the respective spectra of T_1 and T_2.

Thus, by summing these two spectra one can increase the intensity of the even partials if A_1 and A_2 are of the same sign, or decrease it (possibly to zero) if they are of opposite signs. In other words, the set of impulses to use, the sum of the sets 1 and 2, has a $2F0$ frequency and an amplitude that alternates successively between $A_1 + A_2$ and A_1. We have thus, by a simple control of the amplitude of the impulses, control of the even and odd partials. This is the case in each frequency band. (See "The CHANT Program," discussed later.)

Finally, in the case where the parallel filter is used for the processing of a signal, one can derive J signals that will serve as sources for J cells. By storing the original signal in a table and looking it up

with dephased indexes, one creates an equivalent number of dephased sources. One can also introduce other variations (for example, amplitude variations), especially if each fundamental period can be detected with certainty (this can be tricky).

The Need for Floating-Point Arithmetic

We have referred to the difficulties presented by a series filter model. The first difficulty is inherent in the integer-based arithmetic used in most synthesizers. Already, with a single cell, one can easily encounter an auto-oscillating, and therefore unacceptable, filter (this is the case on the Samson Box for a "modifier" used in the two-pole mode). Moreover, from one cell to another, the noise inherent in the calculations increase, and numerical overflows are common.

With some effort, using judiciously calculated filter gains in a specific application, one can usually cure these problems. But then the model loses all its flexibility and generality. A specialist is required to carry out the adjustments and that means a considerable waste of time. Ideally, a computer instrument should be as flexible as possible and not confine the imagination of the musician within a labyrinth of calculations exactly at the point where new musical horizons open up.

The second difficulty is the computationally-intensive nature of the parameter calculations for a series filter, on the basis of data that the user can manipulate, such as frequencies, bandwidths, and amplitudes. These calculations require burdensome operations like division and exponentiation to be computed every 20 msec when the filter varies fairly quickly. Again, these calculations require a range and precision that only floating-point arithmetic can provide.

We do not deduce from this that the series model must be abandoned. Quite the contrary, because we have shown that the parallel model presents its own difficulties. But musical synthesizers must include very rapid floating-point arithmetic and the operations absolutely necessary to any modern computer. Present day technology allows this, in effect, for a reasonable cost that is declining year by year. Moreover, the extra cost (over fixed-point arithmetic) is compensated for by the power and flexibility that the synthesizer gains.

Two other arguments plead in favor of floating-point arithmetic. First, one must correctly adjust the level of the audio signal for digital-to-analog conversion. Who has not suffered from saturated or inaudible sounds and lost a great deal of time in adjusting magical gain coefficients? In a synthesizer working to a large extent in floating-point arithmetic, an "empty" pass is enough to detect the maximum amplitude sample, to deduce the gain and to play (without human intervention) a sound whose level is optimally adjusted. This is the case in the CHANT program: the sound is never saturated or inaudible, and this is a significant factor in its productivity and pleasure.

The second argument relates to the calculation of the parameters, that is, all the calculations that precede the synthesis itself. These algorithms can be very complex, such as when emulating the refinements of traditional interpretation, and mastering all the details of a sound. These algorithms cannot be reduced to a few table lookups and interpolations. Moreover, the user must be free of all concern with the numerical range of the calculations and other overflows (the algorithms and the music are already quite complex enough). Consequently, parameter calculations must also use rapid floating-point arithmetic. We show later that these calculations also require sophisticated software.

With the aid of floating-point arithmetic, the computer can lend itself to the most extreme and unexpected uses. One of the merits of the CHANT program is that it accepts parameters that have, as far as possible, an immediate meaning. Moreover, one merely has to adjust the parameters at values that are not absurd from the physical and perceptual point of view to obtain a sound that is both expected and surprisingly rich.

The CHANT Program

General Description of the Program

The CHANT program was conceived as an interactive instrument. A distinction can be made between two modes of utilization, conditioned both

by the type of sounds that the user desires to synthesize and by the user's experience. The first mode of utilization, which can be described as "basic," corresponds to the context of singing voice synthesis, starting with the basic rules defined in the program. The second mode, which can be described as "extended," corresponds to applications that demand either different or more developed controls than the basic ones implemented in the program, or that move from the vocal model toward other models, other "instruments," or other approaches (for example, additive synthesis).

In the first mode, the user simply specifies the values of the preexisting parameters, in the second mode the user has all the power of a programming language to specify algorithms to describe parameter evolutions, that is, to modify the basic rules or to create new rules.

In the basic version, CHANT includes about a hundred parameters. But many remain unchanged from one sound to another, so all parameters are consequently defined with default values, enabling immediate freedom of use. These parameters can be grouped together under the following headings:

Frequency of the fundamental
Random variations of the fundamental
Vibrato
Random variations of the vibrato
Spectrum: formants and fundamental
Slope of the spectrum
Automatic calculation of the spectrum
Intensity of the sound
Local envelope of the formants
Control over the synthesis

Each parameter can be defined either by a fixed value, that is, by a constant, or by a function of time (breakpoint functions) that associates a given value with a given time and consequently allows interpolation from one value to another.

These parameters and their values are stored in a file known as the *parameter file*. Other files may contain only functions, for example, complex functions calculated with tools in the CHANT environment and defined by a large number of points. The latter, known as *function files* can be called from a main file that describes all the calls.

Thus the data files can be used in modular fashion and constitute a network, sometimes of considerable complexity. In particular, by the interplay of factors and offsets applied to the functions, the files taken from the library can be used as models from which one progressively deviates by successive modifications and by the readjustment of values until the desired effect is obtained.

When CHANT is used in the "basic" mode (mainly voice synthesis), the user's work consists of editing a parameter file, either by directly defining the values or by modifying the values already defined if the user starts with a preexisting library file.

But these modifications can also be effected in an interactive mode at the last minute, right before starting the synthesis. In this case the user gives the program a parameter file and then only modifies particular values. The original file that was used as a model remains intact, but the modifications are preserved in an output file automatically produced by the program. This output file is therefore a replica of the input file, with the addition of the modifications inserted in the interactive mode. Hence, work can be stored and retrieved. The output file can be used as an input and remodified, or else it can be kept to trace each stage of the development. This demonstrates the special care taken to make CHANT an evolving instrument—with memory. These files are kept in a library of parameter and function files. This libary represents the first part of what we have called the CHANT environment, which also includes a catalog of programs and subprograms, consisting of tools for analysis, for function definition, and for rapid spectrum construction, among others.

Knowledge Models

The ability to save accumulated work springs from the same concerns as those that led us to envisage the definition in terms of schemas that are *knowledge models* of a given production. In order to be efficient, the definition of models, which is costly in experimentation time, must lead to knowledge that is easily accessible and can be reused.

This first level of intervention in CHANT could

be called the specification of a *data model*. The data controls the rules, the underlying algorithmic descriptions that form (by default in CHANT) a vocal model. But this standard model can also be modified by additions and/or deletions of rules and algorithms. This constitutes a second level of intervention in CHANT that could be called the specification of a *rule model* that we describe in a moment.

Let us now consider some of the rules specified for the vocal model, from fairly simple rules for the timbre of vowels, to more complex vocal rules that describe the relationship between timbre and amplitude.

Development of the Vocal Model

Specifying the vocal model consisted of precisely defining and realizing the timbres, in particular those of vowels, in the singing voice. This includes the frequency and amplitude of the vibrato, random variations of the fundamental frequency and of the vibrato, and relations between timbre and amplitude. At first, timbre is considered on the basis of a spectral envelope, which is itself defined in terms of center frequencies and amplitudes of the formants.

We represent the spectral envelope by its formants. This representation has shown itself to be particularly economical and informative, both at the level of the competence required for understanding the result of a spectral analysis (either analyzed by machine or by ear) and at the level of performance (synthesis). The formant frequencies were extracted by means of an original analysis with the phase vocoder, that is, by extracting the frequency and amplitude evolutions of each partial and by superimposing them to reveal their correlations and so deduce the formant frequencies. We also used more classical analysis tools offered by linear predictive coding.

Use of the data we developed for our vocal model is optional. One can also choose to have the bandwidths of the formants computed automatically, on the basis of the formant frequencies. This is done by following a parabola defined on the spectrum by three points; these points are themselves adjustable parameters.

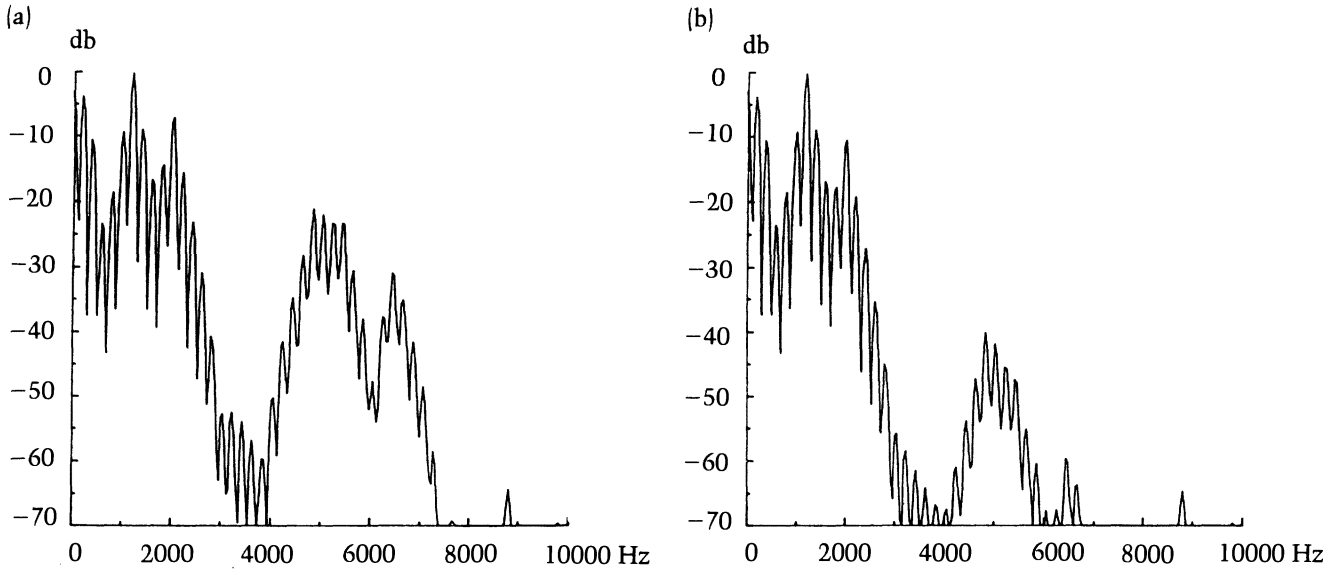
Automatic calculation of the amplitude of the formants can be accomplished by simulating a filter series and, according to the frequency of each filter, fixing their amplitudes. Thus, when two or more formants approach each other their amplitudes are reinforced. A supplementary formant, known as a *complement*, can also be automatically assigned to modify the first formant and give more energy in the low register. This is, in a way, a "zeroth" formant.

Random variations in the fundamental (called *jitter*) are perceptually very important. By asking singers to produce sounds with absolutely no vibrato we were able to study the random and uncontrolled fluctuation of the frequency of the fundamental. In the analysis many irregularities were observed, although the variation is only of the order of $\pm 0.5\%$ from the fundamental frequency. Considerable variations in size were also observed. The random fluctuation of the fundamental follows a distribution close to $1/f$.

In the CHANT program this fluctuation is modeled by adding a term to the frequency of the fundamental. This term varies at random between limits given by the user, and is the sum of three components whose values are obtained by interpolation between independent, periodic random choices of frequency deviation. (Typical values for the three random periodicities are 0.05, 0.111, and 1.219 sec.) We distribute the excursions of the random fluctuations equally between the three components. The total fluctuation is typically situated between 1.1% and 3.7% from the fundamental for women's voices and between 2.0% and 5.7% for male voices.

Vibrato is traditionally defined as the more or less regular oscillation of the fundamental frequency around a center frequency that is perceived as a pitch. In CHANT, we distinguish between the amplitude of the vibrato—that is, deviation around the center frequency—from the frequency of the vibrato—that is, the *repetition rate* of this deviation and random variations of it. Vibrato is interesting from a timbral point of view, and it is important in the recognition of the identity of a singer. It

Fig. 16. (a) Log magnitude spectrum of the vowel /a/ at a high amplitude. (b) The same vowel at a low amplitude.



sweeps across vocal tract resonances corresponding to the formants and, consequently, reveals them to the ear. This is also why vibrato also plays an important role in interpretation.

Special care has also been given to the study of the relation between timbre and amplitude. When a vocalist sings loudly, the signal emitted from the vocal cords is completely different from that produced when the vocalist sings softly. It happens that this loud signal is much richer in high frequencies. The same difference exists between the notes at the top of a register and those at the bottom. We have modeled these effects by applying a corrective function to the amplitudes of the formants number 2–5.

This correction is, on the one hand, a function of the general amplitude demanded by the user, and on the other hand, a function of the position of the note in the register requested by the user. In CHANT, we define a *register* by the frequency that corresponds to the middle of the register desired. Thus, the same vowel synthesized at the same frequency will have different timbres depending on the choice of the frequency that defines the register.

Figures 16a and 16b show the spectrum of two sounds synthesized by the program. The fundamental is 300 Hz, the vowel is /a/. In Fig. 16a, the amplitude is at a maximum; in Fig. 16b it is much lower. The difference in the amplitude of the higher formants is noticeable.

The relation between timbre and amplitude is also perceptually important as an indication of the distance of a sound. If one applies a spectrum correction corresponding to a loud sound, while using a low amplitude for the synthesis, one hears the sound in the distance.

Finally, other controls have been defined to allow such things as tremolo, hoarseness, and balance between the fundamental and the first formant versus the higher formants.

The vocal model in CHANT has been specifically discussed elsewhere, so we will not enter in further details here. See Rodet and Bennett (1980) and Bennett (1981).

Model by Rules

The model just described is implemented in the program so as to answer to typical needs. But it is often necessary to go further towards less common uses. This is why we have designed CHANT to accept input from external subprograms that enable

the definition of new correlations or rules. These subprograms are written directly in the implementation language of CHANT (or in any other available language) and are executed at a low frequency (typically 100 Hz), not at the sampling rate. All the parameters of the program are accessible in a simple manner.

These rules form the basis of the models that, ideally, should be both modular and context-sensitive. To illustrate this point, we present a few examples taken from two subprograms that are already complex. The first is an attempt to simulate a soprano voice with elements of phrasing and classical singing technique, and the second (using a voice inspired by Tibetan chant) includes consonant articulation and some original work on timbre.

Bel Canto Voice

For the soprano voice, we first emphasized the placing of the formants in a manner typical of the production of bel canto. On the basis of analysis performed with the phase vocoder on the same pitch interpreted by several singers, we were able to obtain precisely the frequencies of the first eight formants. Ultimately, we decided to keep the frequencies of the last six formants, and for the first two we revealed the following relationship between the frequencies of the formants and the pitch of the note. The first and the second formants are placed respectively on the first and second harmonics, except when the frequency so obtained is below a threshold fixed for both of them. This model produces a homogeneous vocal color over a large tessitura of about two octaves.

At this point, the task of modeling was concerned with the establishment of rules constraining the evolution of the various parameters of the source during staccatos.

In particular, these rules are concerned with describing the evolution of the following parameters:

- Average pitch, described by the shape of an internal portamento
- Vibrato, described by the increase and decrease of the pitch's amplitude and frequency
- Energy, described by an envelope composed of three successive sinusoidal arches
- Vocal effort, described linearly with the amplitude during the attack and the fall of a note only. (In the body of a note the effort continues to grow although the amplitude has stabilized.)

Tibetan Chant

In the work inspired by Tibetan chant, our main concern was to move away from the study of the conventional practices of Western music. Actually, until then the task of defining rules had mainly been concerned with bel canto, on voices which could be described as "trained," that is, tending to eliminate or to "regionalize" noise and randomness, except those expressive qualities that are very specific to Western music.

In the Tibetan chant work, we have emphasized several factors: the structure of a certain type of noise, separate control of the even and odd harmonics, and especially articulation, that is, consonant articulation. In the regular CHANT program, noise is controlled by formant-dependent parameters. That is, one sets up a noise bandwidth centered on the frequency of a formant (in this case a filter) and a noise amplitude. But in this example, noise has been approached by working mainly on random aspects, especially at the level of microfluctuations of the fundamental (different from jitter) and the frequencies of the formants.

For the timbre of the chanting, we have introduced the idea of another coefficient that provides separate amplitude controls for the even and odd harmonics, each controlled by an envelope and a random variation. This coefficient enables one to play with the *roughness* of the sound (already existing in the basic functioning of the program in another way) and also enables one to play with *fusion* and *fission* of the auditory image (McAdams 1984).

Articulation has been worked on intensively. Consonants have been modeled and constructed in the form of *transitions* from one vowel to another, affecting the amplitude, the fundamental, and the *formant trajectories*, that is, the frequency of each formant as a function of time.

Finally, rhythm and stress have been determined by rules describing the correlations between the length of the phonemes and local variations of vo-

cal effort, fundamental frequency, and vibrato. This did not present any special difficulty once the definition of a transition was derived, apart from making CHANT more unwieldy to use. These rules are computationally inefficient and thus are not easy to use in musical applications. This type of difficulty suggests special procedures that we describe later.

A number of rules have also been defined that illustrate the power of the formant representation. In the practice of using CHANT, the formant representation suggested interesting ideas by offering specific cues or access to timbre control (Barrière 1983) and sound imaging (McAdams, forthcoming). These include the following:

- Timbre fusion and fission by stretching or compressing formant frequencies and amplitudes
- Granulation by different layers of random variations on formant frequencies, bandwidths, and amplitudes
- Fusion and fission by playing between superimposition and remoteness of the FOF formants and of the noise formants (filters)
- Spectral enrichment modulation, by different asynchronous levels of control of a curve that increases and decreases formant amplitudes
- Reduction of formants to partials, by moving from a formant as a set of harmonics to a partial centered at the center frequency of the formant (crossing from harmonic to inharmonic)
- Fusion and fission by concentrating or dispersing the formants of a spectral image into space
- Transformation patterns between spectral envelopes by controlling the modes of transition/interpolation between their formants
- Correlations between the abovementioned rules by several hierarchical levels of envelopes

Many of these rules have been generalized and are used extensively.

Models and Derivations

Our proposal, at what one could call the highest level, is therefore to construct models of laws and rules for all stages of musical production, from precomposition, to the choice of sonic materials, to performance. These models, in the form of data and algorithms, must describe as precisely as possible the sound and its evolution, as well as the structure in which it is placed and the dynamic interactions between sound and structure—between sonic material and organization. The idea of these models is not simply to imitate or simulate a given note or instrument in a given context, but rather to represent the formalization of any act or process, of a decision or a gesture, of a static or dynamic organization, or of a musical invariant in general.

These models must be viewed as *knowledge structures* or *knowledge schemes*, from which one deviates by successive modification or composition. They are in a way propositions in the quasi-logical sense of the term, and, consequently composers are not obliged to see their structure in depth. Composers can often content themselves with manipulating and assembling them, as they have always done in instrumental or vocal music. Thus, an understanding of the internal workings must not be made an essential prerequisite, but rather an optional endeavor (but not an occult science). In all cases, flexibility must be preserved.

Of course, it is not enough to be able to do everything in theory, it must be possible to do it simply and without having to rewrite everything for each application. In particular, a transition must be realized by a preexisting object that one manipulates symbolically and places "on" another object, or between two objects. This quasi-modular or symbolic processing derives almost automatically from the conception of models such as we have described. Thus, a transition is also in this sense a model to be placed in a context, without concern for the compatibility between two or several objects.

In the CHANT project, once the specific task of modeling was carried out, we then saw a need for a structure at the highest level to manage the models as objects or processes in a symbolic combination. This would enable both reciprocal modifications of the objects/processes "in context," and control at a still higher level.

These are the imperatives that led us to envisage a new program and a new language to control CHANT and other systems of synthesis. This program, known as FORMES (Rodet and Cointe 1984)

and implemented in the Lisp language, deals directly with problems relating to artificial intelligence. In particular, it deals with structures of knowledge, of constraints, and of parallelisms and transformations, that is, the algorithmic aspects of rule sequences, both at the horizontal level of temporal succession as well as at the vertical level of rule specification and combination.

FORMES provides a framework to manipulate and integrate models and rules as basic building blocks or functional objects with scheduling characteristics. FORMES is both an answer to questions arising in the course of CHANT's evolution, and a new direction evolving beyond these questions. CHANT extended the composition process and provided powerful tools for the composition of sonic material. FORMES starts where CHANT stops, in an attempt to process—with one set of tools—both synthesis and composition problems: sonic material and musical organization.

Conclusion

The CHANT program starts from but exceeds the study of vocal behavior. In any case, we do not consider the voice as a simple, single-faceted object. On the contrary, we have taken the voice as a starting point because of its richness and complexity. The issue at stake is principally musical but at the same time cognitive, as, in our opinion, the two aspects are intimately linked, particularly in the modeling process.

The sonic quality of the synthesis and the ease of use of the program make CHANT an exceptional instrument for computer music. Since its first implementation, it has been used in a wide variety of musical contexts by numerous composers, including Gerald Bennett (1981), Conrad Cummings, Jean-Baptiste Barrière (1983), Jonathan Harvey (1981; 1984), Jukka Tiensuu, Harrison Birtwistle, Kaija Saariaho (1983), Michel Tabachnik, Gerard Grisey, Alejandro Vinao, Tod Machover, and Marco Stroppa. A large number of models have been defined and used, taking special care at first to encompass all the traditional instruments. We have synthesized very good strings (contrabasses, violins, cellos), winds (trumpets, oboes, clarinets, horns, flutes), percussion (drums, cymbals, gongs, gamelan, bells), and other instruments.

The definition of these models has allowed composers to place special emphasis on timbre, for example by defining imaginary hybrid instruments, or sophisticated interpolations between points in a timbre space. The synthesis-by-rule approach has also facilitated investigation into sounds that are quite removed from instrumental references, such as additive synthesis textures and inharmonic sound synthesis.

Although in permanent use at IRCAM, CHANT continues to be developed, enriched by new uses and new implementations on different machines. The first version of CHANT was written in the Sail language for a DEC PDP-10 by Xavier Rodet, Yves Potard, and Conrad Cummings at IRCAM and ran between 1979 and 1983. A portable version in Fortran was written in 1981 by Jean Holleville. This same Fortran version was ported from the PDP-10 to a DEC VAX-11/780 running the Unix operating system in 1983.

The entire library of user subprograms has since been translated into both Fortran and the C language. The rule-based knowledge embedded in these subprograms was transferred into FORMES (Rodet and Cointe 1984) by Jean-Baptiste Barrière and Xavier Rodet so that this knowledge base can also be used outside of CHANT with other synthesis devices.

With the help and support of John Gordon and John Chowning, CHANT has been running since 1981 in its filter version on the Systems Concepts digital synthesizer at CCRMA, Stanford University. The Fortran version is also running on a VAX-11/750 at the Electronic Music Studio in Stockholm. In 1983, a real-time filter version of CHANT was implemented by Xavier Rodet and Yves Potard on the 4X real-time digital sound processor at IRCAM.

Finally, in 1984 Yves Potard implemented a new version of CHANT on a Floating Point Systems FPS-100 Array Processor. This very fast implementation provides a combination of the FOF version and the filter version. Moreover, the sound source can be derived externally—for example, concrète sounds can be used. This makes possible processing

and even cross-synthesis by combining the two models. CHANT has therefore, today, become a complete package of synthesis and processing.

References

Barrière, J.-B. 1983. "Chreode I: A Piece Using CHANT and FORMES." Presented at the International Computer Music Conference, October 1983, Rochester, New York.

Bennett, G. 1981. "Singing Synthesis in Electronic Music." In *Research Aspects of Singing*, ed. J. Sundberg. Publication 33. Stockholm: Royal Swedish Academy of Music, pp. 34–50.

Carré, R. 1981. "Couplage conduit vocal—source vocale." In *XIIème Journées d'études sur la parole, Montreal, May*, pp. 233–245.

Fant, G. 1970. *The Acoustic Theory of Speech Production*. The Hague: Mouton.

Fant, G. 1973. *Speech Sounds and Features*. Cambridge, Massachusetts: MIT Press.

Harvey, J. 1981. "Mortuos Plango, Vivos Voco: A Realization at IRCAM." *Computer Music Journal* 5(4):22–24.

Harvey, J. et al. 1984. "Notes on the Realization of Bhakti." *Computer Music Journal* 8(3):74–78.

McAdams, S. Forthcoming. "The Auditory Image: A Metaphor for Physical and Psychological Research on Auditory Organizations." In *Cognitive Processes in the Perception of Art*, ed. R. Crozier and A. Chapman. Amsterdam: North-Holland Publishers.

Moorer, J. A. 1977. "Signal Processing Aspects of Computer Music: A Survey." *Proceedings of the IEEE* 65(8):1108–1137. Reprinted in *Computer Music Journal* 1(1):4–37.

Rodet, X. 1977. "Analyse du signal vocal dans sa representation amplitude-temps, Synthèse de la parole par règles." Thèse d'Etat. Université de Paris VI. Paris.

Rodet, X., and G. Bennett. 1980. "Synthèse de la voix chantée par ordinateur." In *Conferences des Journées d'études 1980*. Paris: Festival International du Son, pp. 73–91.

Rodet, X., and P. Cointe. 1984. "FORMES: Composition and Scheduling of Process." *Computer Music Journal* 8(3):32–50.

Rothenberg, J. 1981. "The Voice Source in Singing." In *Research Aspects of Singing*, ed. J. Sundberg. Publication 33. Stockholm: Royal Swedish Academy of Music, pp. 15–33.

Saariaho, K. 1983. "Using the Computer in a Search for New Aspects of Timbre Organization and Composition." Presented at the International Computer Music Conference, October 1983, Rochester, New York.

Sundberg, J. 1978. "Synthesis of Singing." *Swedish Journal of Musicology* 60(1):107–112.

Sundberg, J. 1979. "Perception of Singing." Speech Transmission Laboratory Quarterly Progress and Status Report 1-1979. Stockholm: KTH, pp. 1–48.

Weinreich, G. 1977. "Coupled Piano Strings." *Journal of the Acoustical Society of America* 62(6):1474–1484. Also published in 1979 as "Physics of Piano Strings." *Scientific American* 240:117–127.

Discography

Barrière, J.-B., ed. 1983. *IRCAM: un portrait—recherche et création*. IRCAM 0001. Paris: IRCAM.