

# Um Algoritmo de Síntese FM Controlada por Caos

*Edson S. Zampronha*

## Introdução

O século XX passou a tratar o timbre como um dos elementos essenciais da construção musical (Barrière 1991). As músicas concreta e eletrônica, no meio do século XX, fizeram uso de diferentes processos de gravação para a obtenção de novos timbres. No final da década de 1950 essas duas estéticas musicais, embora distintas entre si, passaram a ser denominadas música eletroacústica (Caesar 1997), explorando tanto a síntese quanto o tratamento sonoros. Também no final da década de 1950 foi produzido, pela primeira vez, um som digital inteiramente realizado em um computador (para uma visão histórica ver Schwanauer & Levitt 1993).

Os primeiros procedimentos de síntese sonora, também da década de 1950 (para uma visão geral sobre diferentes procedimentos de síntese ver Roads 1994), partiam do princípio de que timbres simples, periódicos, com altura definida, possuíam uma dinâmica simples. Portanto, supunha-se que também deveriam possuir um modelo simples que os gerassem. Inversamente, supunha-se que timbres complexos, com grande riqueza espectral tendendo ao ruído, possuíam uma dinâmica complexa, e portanto deveriam necessitar de um modelo complexo para serem gerados. Por esta razão as primeiras tentativas de geração de timbres complexos utilizaram como ferramenta básica os recursos estatísticos (Xenakis 1971). Na maioria dos casos neste tipo de síntese “não há um modelo acústico preexistente” (Di Scipio & Prignano 1994, 15).

No entanto, há algumas décadas começou-se a observar que a maioria dos sistemas complexos na natureza, que antes se supunha só poderem ser descritos com métodos estatísticos, na verdade resultavam de modelos simples, deterministas, embora imprevisíveis. A isto se denominou *caos* (Sole & Manrubia 2000, p.147-8). O com-

portamento caótico se encontra em uma estreita faixa entre comportamentos periódicos (ou semi-periódicos) e comportamentos ruidosos. Este comportamento caótico pode ser observado em diversos fenômenos naturais (Nicolis e Prigogine 1989), incluindo fenômenos sonoros. Para a obtenção de sons com estas características, no entanto, não é necessária a criação de um novo modelo de síntese. Este resultado pode ser obtido através de novos modos de utilização de sínteses tradicionais, como síntese aditiva e subtrativa ou síntese FM, entre outras.

O propósito deste trabalho é mostrar como é possível a utilização de modelos caóticos para o controle de procedimentos de síntese tradicionais de modo a se conseguir sons com comportamento caótico. Apresenta-se um algoritmo no qual o modelo caótico a ser utilizado provém da equação logística, e o procedimento de síntese a ser controlado é a síntese FM. Este algoritmo é apresentado em BASIC, e gera como resultados os arquivos necessários para que a síntese FM possa ser realizada em CSound. Mostra-se ainda como este algoritmo pode ser modificado de forma a se conseguir outros resultados.

## **O modelo caótico: a Equação Logística**

Um comportamento complexo caótico pode ser obtido através do uso de equações simples, desde que não lineares. Uma equação deste tipo, bastante conhecida, é a equação logística. Sua apresentação, no presente trabalho, será dirigida a mostrar seu uso no controle da síntese FM. Um estudo pormenorizado da equação logística pode ser encontrado em Ferrara & Prado (1994, p.109-124). A equação logística possui a seguinte forma:

$$X_{n+1} = RX_n(1-X_n)$$

Esta equação produz uma seqüência de números. Dado um número inicial qualquer, este número é introduzido no lado direito da equação, no lugar de  $X_n$ . Como resultado da operação obtém-se  $X_{n+1}$ , que é o segundo número da seqüência. Este segundo número é introduzido novamente no lado direito da equação e, como resultado, ob-

tém-se o terceiro número da seqüência, e assim por diante. Esta operação pode ser repetida indefinidamente. Há uma instabilidade inicial presente nas primeiras iterações que rapidamente desaparece, levando a equação a se estabilizar em um dos comportamentos descritos a seguir, dependendo do valor de  $R$ . Para efeito de cálculo é importante verificar que o resultado da equação logística, após sua estabilização, está sempre entre 0 e 1.

$R$  é um parâmetro de controle. Ele funciona de modo similar a um botão em um equipamento, que pode ser aumentado ou diminuído. Se  $R \leq 3$ , após as primeiras iterações a equação tende a se estabilizar, e passa a dar como resultado um único valor que se repete indefinidamente. Mas, se  $R$  está entre 3 e 3.5699456... observa-se um interessante fenômeno de bifurcações. Se  $R$  está próximo a 3 observa-se que a equação se estabiliza mas não em um, e sim em dois valores que se alternam indefinidamente. Quanto mais  $R$  se aproxima de 3.5699456... maior a quantidade de números em que a equação se estabiliza. Inicialmente em 2 valores, passa a 4, a 8, a 16, a 32, a  $2^n$ , tendendo ao infinito.

No entanto, quando  $R$  é maior que 3.5699456... e menor que 4 temos o surgimento de um comportamento caótico propriamente dito. Aqui certas regiões são visitadas periodicamente, no entanto os valores efetivamente obtidos dentro de cada região não se repetem. Esta região é chamada de cascata inversa, e quanto mais  $R$  se aproxima de 4 mais estas regiões se fundem, e maior a caoticidade dos resultados obtidos. No entanto a caoticidade não cresce linearmente. Há momentos de periodicidade. Finalmente, quando  $R > 4$  as soluções tendem a  $-\infty$  (menos infinito).

A Figura 1 apresenta diferentes soluções da equação logística para diferentes valores de  $R$ . Observa-se a região de bifurcações e a região caótica com momentos de periodicidade. A Figura 2 ilustra esquematicamente a região de bifurcações e a região caótica com sua estrutura de cascata inversa que se funde conforme  $R$  se aproxima de 4.

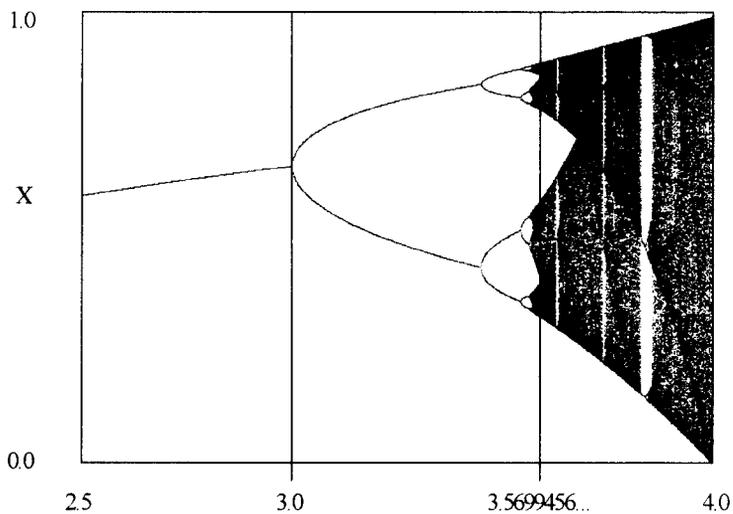


FIGURA 1

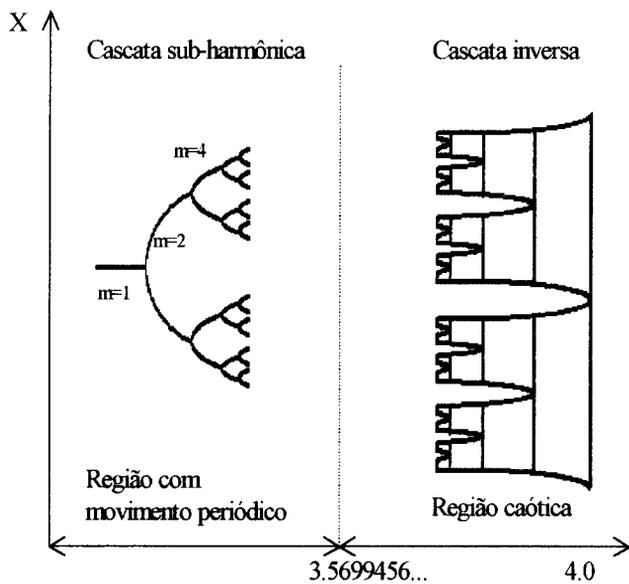


FIGURA 2. Extraído de Fiedler-Ferrara & Prado (1994, p.121, gráfico adaptado)

## A ligação entre a equação logística e a síntese FM

A maior dificuldade na relação entre modelos matemáticos e música é justamente a forma com que se passa de um domínio a outro. Este é realmente o ponto mais delicado, e diversas tentativas se tornam ineficientes justamente aí. Uma tradução direta do domínio matemático no musical, associando os números obtidos a notas, por exemplo, resulta ingênuo já que o próprio conceito de nota é uma abstração (Francès 1984). Um estudo detido sobre representação é necessário, mas transcende o propósito deste trabalho (para um estudo sobre representação em música ver Zampronha 2000a).

Aqui será mostrada uma das formas pelas quais a equação logística pode ser utilizada para controlar uma síntese FM (uma visão pormenorizada sobre síntese FM pode ser vista em Chowning & Bristow 1986). Uma síntese FM simples pode ser realizada com dois osciladores, um com frequência fixa e outro com frequência variando no tempo. A Figura 3 ilustra esquematicamente o processo

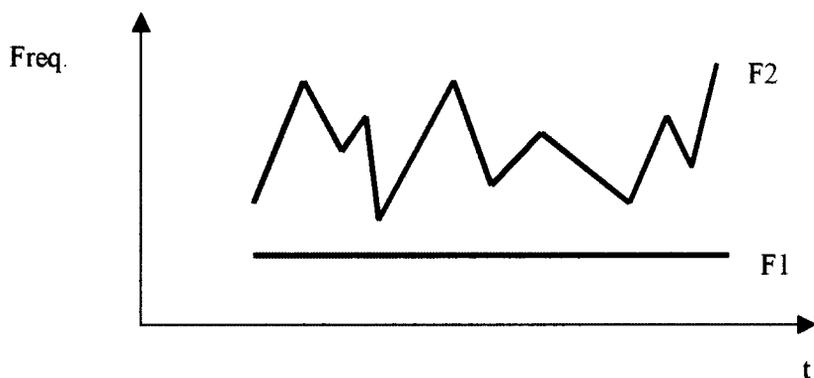


FIGURA 3

Uma frequência fixa, F1, é modulada por outra frequência variável, F2. A lista de números obtida através da equação logística servirá para controlar exatamente as variações de F2. A opção foi fazer F2 variar no campo das frequências realizando *glissandos*. Como a

equação logística oferece uma única série de números, esta série de números deveria controlar apenas um parâmetro na síntese FM, sendo este parâmetro justamente cada frequência a ser atingida por cada *glissando*. No entanto, pode-se utilizar esta equação para controlar dois parâmetros: a frequência e a duração de cada *glissando*. Esta opção foi realizada para se obter um resultado mais complexo sem a necessidade de se utilizar uma equação de duas ou mais dimensões. Uma equação de duas dimensões pode ser, por exemplo, o sistema de Hénon (Sole & Manrubia 2000, p.165-8), que gera duas seqüências de valores que podem ser utilizadas, uma para a frequência e outra para a duração. É possível também a utilização de duas equações logísticas, inclusive com caoticidades diferentes, para controlar a frequência e a duração.

O programa mais acessível para a geração deste tipo de síntese é o CSound. No entanto, como a quantidade de informações a serem digitadas é muito grande e tarda muito, esta solução torna-se inviável quando se quer fazer muitas experimentações. Optou-se por elaborar um programa em BASIC para a realização dos cálculos e para a geração dos resultados em dois arquivos texto, .ORC e .SCO, que são os arquivos lidos pelo CSound. Deste modo os dois programas funcionam em seqüência, e os experimentos podem ser realizados rapidamente.

## **Apresentação do algoritmo**

As linguagens BASIC e CSound estão disponíveis para diferentes plataformas, e são de fácil acesso. Um estudo detalhado sobre CSound com muitos exemplos de utilização pode ser encontrado em Boulanger (2000). A literatura em BASIC é farta e bastante disponível. O algoritmo em BASIC pode ser facilmente implementado em diferentes versões como o VisualBasic, QuickBasic, ou mesmo BASICs mais antigos que, nesse caso, podem requerer numeração de linha. O presente algoritmo foi realizado em plataforma PC e possui comentários que explicam diversas de suas passagens, de tal modo a facilitar eventuais adaptações.

\*\*\*\*\*  
 \*\*\*\*\*  
 Síntese FM controlada por  
 equação logística  
 \*\*\*\*\*  
 \*\*\*\*\*

‘A Equação logística controla a frequência e a duração da síntese FM  
 ‘Escreve o programa em arquivos .orc e .sco para posterior geração em CSound.  
 ‘O resultado é um material bruto a ser tratado posteriormente.  
 ‘Os arquivos dever ser salvos no diretório CSound. Os diretórios aqui são  
 ‘hipotéticos.

‘Parâmetros Iniciais

EqLog = .5 ‘Número de partida para os cálculos da Eq. Logística.  
 R = 3.859 ‘Controla o comportamento da equação.  
 FreqF1 = 600 ‘Frequência da onda F1, em Hz.  
 FreqDeBaseF2 = 1750 ‘Frequência mais grave do glissando, em Hz.  
 FreqBandaF2 = 770 ‘Banda de variação do glissando, acima da  
 ‘FreqDeBaseF2. No presente caso o glissando  
 ‘varia entre 1750Hz e 2520Hz.  
 TempoDeBaseF2 = .07 ‘Menor duração possível do glissando (em segundos).  
 ‘Valores muito baixos provocam erro de interpretação  
 ‘no CSound.  
 TempoBandaF2 = .1 ‘Banda de variação temporal do glissando.  
 ‘No presente caso a duração do glissando varia  
 ‘caoticamente de .07" a .17".  
 AmpF1 = 80 ‘Amplitude da onda f1, em dB.  
 AmpF2 = 34.6 ‘Amplitude da onda f2, em dB.  
 NumIteracoes = 7 ‘Número de parâmetros por linha de comando. Valores  
 ‘acima de 60 podem dar “estouro de pilha” no  
 ‘CSound. O valor mínimo deve ser superior a 2.  
 DurTot = 7 ‘Duração total do som a ser gerado (em segundos). O  
 ‘programa não trunca a geração sonora. Se o tempo de  
 ‘geração, resultado de NumIteracoes, ultrapassar  
 ‘o valor de DurTot, a duração final será ligeiramente  
 ‘maior.  
 DIM Sco\$(300)  
 DIM tempTot(300)  
 DIM tempSub(300)

‘Gera os 150 primeiros números da equação logística para excluir qualquer  
 ‘traço da instabilidade inicial. Caso se deseje manter estes traços iniciais  
 ‘basta eliminar este FOR - NEXT

```

FOR y% = 1 TO 150
  EqLog = R * EqLog * (1 - EqLog)
NEXT

```

‘Gera informações para a elaboração do arquivo .ORC, do CSound. Aqui são  
‘definidos quantos parâmetros terá cada linha de comando do CSound, em função  
‘do valor de NumIteracoes.

```

cont = 0
Orq$ = ""
FOR x = 0 TO NumIteracoes
  cont = cont + 1
  Orq$ = Orq$ + "p" + STR$(2 + cont * 2)
  IF x < NumIteracoes THEN
    Orq$ = Orq$ + "," + "p" + STR$(3 + cont * 2) + ","
  END IF
NEXT

```

‘Gera lista de parâmetros (frequência e duração) a ser usada do arquivo .SCO  
‘para controlar os glissandos.

```

cont = 0
tempoTotal = 0
DO UNTIL tempoTotal > DurTot
  cont = cont + 1
  FOR x = 0 TO NumIteracoes

    ‘escreve frequência
    IF x = 0 AND cont > 1 THEN
      Sco$(cont) = Sco$(cont) + STR$(ft1)
    ELSE
      IF x = 1 AND cont > 1 THEN
        Sco$(cont) = Sco$(cont) + STR$(ft2)
      ELSE
        ft1 = freq
        EqLog = R * EqLog * (1 - EqLog)
        freq=INT(FreqDeBaseF2+FreqBandaF2 * ((10 ^ EqLog) / 10))
        Sco$(cont) = Sco$(cont) + STR$(freq)
        ft2 = freq
      END IF
    END IF

    ‘escreve duração
    EqLog = R * EqLog * (1 - EqLog)
    IF x = 0 AND cont > 1 THEN

```

```

    Sco$(cont) = Sco$(cont) + STR$(tt1)
    tempTot(cont) = tempTot(cont) + temp
ELSE
    IF x < NumIteracoes THEN
        temp = TempoDeBaseF2 + TempoBandaF2 * EqLog
        temp = INT(temp * 1000) / 1000
        Sco$(cont) = Sco$(cont) + STR$(temp)
        tempTot(cont) = tempTot(cont) + temp
        tt1 = temp
    END IF
END IF
NEXT
tempSub(cont) = temp
tempoTotal = tempoTotal + tempTot(cont) - tempSub(cont - 1)
LOOP

```

‘Escreve cabeçalho do arquivo .ORC. Este cabeçalho pode ser alterado conforme ‘a resolução que se deseje obter. Todos os arquivos são direcionados para ‘serem salvos no diretório onde está o CSound. O diretório aqui indicado é ‘apenas um exemplo.

```

OPEN "c:\Csound\caosFM.orc" FOR OUTPUT AS #1
PRINT #1, "sr = 44100"
PRINT #1, "kr = 4410"
PRINT #1, "ksmps = 10"
PRINT #1, "instr 10"
PRINT #1, " iamp1   =   ampdb(" + STR$(AmpF1) + ")
PRINT #1, " iamp2   =   ampdb(" + STR$(AmpF2) + ")
txttemp$ = " kenvelope linseg 0, p5, 1, p3-p5-p"
txttemp$ = txttemp$ + STR$(NumIteracoes * 2 + 3) + ", 1, p"
PRINT #1, txttemp$ + STR$(NumIteracoes * 2 + 3) + ", 0"
PRINT #1, " ksweep   linseg " + Orq$
PRINT #1, " asig2    oscil iamp2, ksweep, 1"
PRINT #1, " asig1    oscil iamp1," + STR$(FreqF1) + " *asig2, 1"
PRINT #1, "          out  asig1*kenvelope"
PRINT #1, "endin"
CLOSE #1

```

‘Escreve cabeçalho do arquivo .SCO. O diretório indicado é apenas um exemplo.

‘Os parâmetros em cada linha de comando estão escritos na seguinte ordem:

‘instrumento início duração amplitude freq dur freq dur freq, etc.

‘O comando “f1 0 4096 10 1” define a forma de onda como uma senóide.

```

OPEN "c:\csound\caosFM.sco" FOR OUTPUT AS #2

```

```

PRINT #2, " f1 0 4096 10 1"
FOR y% = 1 TO cont
  temp = 0
  FOR x% = 1 TO y%
    temp = temp + tempTot(x% - 1) - tempSub(x% - 1)
  NEXT
  txttemp$ = "i10" + STR$(INT(temp * 1000) / 1000)
  PRINT #2, txttemp$ + STR$(tempTot(y%)) + Sco$(y%)
NEXT
PRINT #2, "e"
CLOSE #2

```

‘Comando para executar o Csound de dentro do próprio programa em BASIC.  
 ‘A sintaxe para que o CSound opere a partir dos arquivos ORC e SCO é “csound  
 ‘caosFM.orc caosFM.sco”. No entanto, é recomendável que os caminhos  
 ‘completos que indicam onde estão os arquivos sejam indicados.  
 CLS  
 SHELL “c:\csound\csound c:\csound\caosFM.orc c:\csound\caosFM.sco”

‘Conclui o programa  
 END

Como ilustração, apresenta-se os arquivos caosFM.orc e caosFM.sco gerados pelo programa acima:

### *CaosFM.orc*

---

```

sr = 44100
kr = 4410
ksmps = 10
instr 10
iamp1      =      ampdb( 80)
iamp2      =      ampdb( 34.6)
kenvelope linseg 0,p5,1,p3-p5-p17,1,p17,0
ksweep linseg p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18
asig2      oscil  iamp2, ksweep, 1
asig1      oscil  iamp1, 600*asig2, 1
           out    asig1*kenvelope
endin

```

---

```
f1 0 4096 10 1
i10 0 .9900001 2147 .149 2085 .159 1933 .16 1914 .155 1986 .166
1854 .114 2442 .087 2020
i10 .903 .8290001 2442 .087 2020 .121 2458 .083 1967 .165 1862 .123
2453 .084 1982 .166 1855
i10 1.566 .9 1982 .166 1855 .085 2003 .166 1855 .115 2445 .086 2007
.166 1856 .116 2452
i10 2.35 .855 1856 .116 2452 .166 1854 .114 2441 .087 2023 .165
1862 .123 2453 .084 1983
i10 3.121 .817 2453 .084 1983 .115 2446 .086 2005 .166 1855 .115
2448 .085 1999 .166 1854
i10 3.772 .905 1999 .166 1854 .087 2024 .165 1863 .123 2449 .085
1993 .166 1854 .113 2437
i10 4.564 .88 1854 .113 2437 .164 1871 .131 2382 .1 2249 .128 2411
.093 2133 .151 2044
i10 5.293 .861 2133 .151 2044 .137 2298 .118 2458 .083 1966 .165
1862 .123 2452 .084 1984
i10 6.07 .816 2452 .084 1984 .115 2444 .086 2011 .166 1857 .117
2456 .083 1972 .165 1859
i10 6.721 .938 1972 .165 1859 .083 1962 .165 1866 .126 2430 .089
2063 .161 1900 .149 2075
e
```

---

## Algumas possibilidades de modificação do algoritmo

São apresentadas aqui algumas possibilidades de modificação do algoritmo visando a obtenção de diferentes resultados. Inicialmente observa-se que a síntese FM está sendo realizada com senóides. Deve-se considerar a possibilidade de realiza-la com outras formas de onda. Há a possibilidade, também, de se fazer com que F1 ou F2 seja um som sampleado, sendo possível que os dois sejam sampleados. Deve-se observar, neste caso, que o resultado tende a ser muito complexo, dependendo da natureza dos sons utilizados. A introdução de sons sampleados, neste caso, é feita através da função *loscil*, do CSound.

O Csound possui uma função interna, chamada *foscil*, que automaticamente realiza uma síntese FM. No presente caso esta função não foi utilizada justamente para que se pudesse conseguir uma grande possibilidade de variações de controle pela equação logística. A relação de amplitude entre as duas ondas é um elemento decisivo na obtenção dos resultados. Esta relação é muito sensível, e pequenas transformações geram resultados muito distintos. Além dos parâmetros

frequência e duração é possível controlar também o parâmetro amplitude de um ou dos dois sons utilizados. Deste modo o espectro estará variando caoticamente no tempo.

O glissando é realizado de forma linear, através do comando *linseg*. Caso se deseje que o glissando seja realizado de forma exponencial este comando deve ser modificado para *expseg*. O número de parâmetros que segue ao comando *linseg* (ou *expseg*) está em função da variável *NumIteracoes*. Como *linseg* (ou *expseg*) é chamada diversas vezes no programa, para que não haja “clicks” sempre que a função for novamente chamada ela realiza um *crossfade* com a chamada anterior. Diminuindo-se o valor de *NumIteracoes* aumenta-se o número de *crossfades*. O valor de *NumIteracoes* deve ser maior ou igual a 2. *NumIteracoes* com valores muito altos (acima de 60 aproximadamente) provocam “estouro de pilha” no CSound. Como o *crossfade* foi pensado para o comando *linseg*, com *expseg* ele ficará muito perceptível. Uma solução é adotar *NumIteracoes* com valor alto para diminuir a ocorrência de *crossfades*.

## Considerações finais

Este artigo apresenta um algoritmo que utiliza um modelo caótico extraído da equação logística para controlar uma síntese FM. A equação logística é uma entre várias equações que podem ser utilizadas. É possível pensar-se que o número de parâmetros controlados na síntese deva coincidir com o número de dimensões da equação (ou do sistema de equações). No entanto isso não é obrigatório, já que o resultado buscado deve atender mais às necessidades estético-musicais do que às de uma demonstração matemática.

Tal como em qualquer campo experimental, deve-se considerar que uma boa dose de experiências com o algoritmo deve ser realizada para a obtenção dos resultados musicalmente desejados. Como este algoritmo visa o controle da síntese FM, é recomendável um conhecimento mais detalhado do funcionamento dessa síntese para que a experimentação seja mais rica. Dentre vários trabalhos sobre síntese FM, um trabalho bastante didático e completo é o de Chowning & Bristow (1986).

O resultado sonoro deste algoritmo é concebido como um material bruto, que pode ser tratado por outros recursos de manipulação de áudio, geralmente revelando materiais sonoros de grande valor musical. Operações como *chorus*, *reverb*, compressão, mas principalmente filtragem podem revelar, e geralmente revelam, materiais muito ricos para a escuta. Em outro artigo (Zampronha 2001) tive a oportunidade de mostrar como filtragem e convolução juntas podem dar a estes sons uma forma de tratamento bastante original, que tive a oportunidade de empregar na composição *Trama Nudo Flujo*.

Há um grande campo de experimentações ainda por ser realizado no que se refere à utilização de caos no controle de processos de síntese sonora. É possível a utilização de fractais, autômatas celulares, diversas formas de atratores, procedimentos que envolvam redes neurais, enfim, diversas formas de complexidade visando a geração de novos sons.

Após o impacto e popularização que caos, complexidade e fractais geraram em meados da década de 1980 e início da década de 1990, vive-se agora uma nova fase na qual o entusiasmo foi substituído por uma exploração mais profunda da relação destes temas com outras áreas. No caso da música, a busca por um modo caótico e automático de geração musical está sendo cada vez mais substituída pela utilização destas ferramentas como um instrumento a mais para a geração de sons com interesse estético. Estes estudos têm, inclusive, levado ao surgimento de novas hipóteses com respeito ao funcionamento da escuta (Zampronha 2000b). Esse é um campo de exploração inicial na qual há uma ampla gama de exploração artística ainda por ser realizada.

## **Bibliografia**

- BARRIÈRE, Jean-Baptiste (1991). *Le Timbre, Métaphore pour la Composition*. Paris: IRCAM/Christian Bourgois.
- BOULANGER, Richard C. (2000). *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming*. Cambridge (Massachusetts): MIT.

- CAESAR, Rodolfo (1997). Material e Forma na Música Eletroacústica. In: *Pesquisa e Música*, v.3, nº1, p.4-10.
- CHOWNING, John; BRISTOW, David (1986). *FM Theory and Applications*. Tokyo: Yamaha Music Foundation.
- DI SCIPIO, Agostino; PRIGNANO, Ignazio (1994). The Synthesis of Complex Sonic Events by Functional Iterations. In: I Simpósio de Computação e Música, 1994, Minas Gerais. *Anais...*, Minas Gerais: Universidade Federal de Minas Gerais (UFMG), 1994, p.15-20.
- FIEDLER-FERRARA, Nelson; PRADO, Carmen P. C. do (1994). *Caos*. São Paulo: Edgard Blücher.
- FRANCÈS, R. (1984). *La Percepcion de la Musique*. Paris: J. VRIN.
- NICOLIS, Gregóire; PRIGOGINE, Ilya (1989). *Exploring Complexity*. New York: W.H.Freeman.
- ROADS, Curtis (1994). *Computer Music Tutorial*. Cambridge (Massachusetts): MIT.
- SCHWANAUER, Stephan M.; LEVITT, David A. (Eds.) (1993). *Machine Models of Music*. Cambridge (Massachusetts): MIT.
- SOLÉ, Ricard V.; MANRUBIA, Susanna C. (2000). *Orden y Caos em Sistemas Complejos*. Barcelona: Edicions UPC.
- XENAKIS, Iannis (1971). *Formalized Music*. Bloomington: Indiana University.
- ZAMPRONHA, Edson S. (2000a). *Notação, Representação e Composição: um Novo Paradigma da Escritura Musical*. São Paulo: Annablume/FAPESP.
- \_\_\_\_\_ (2000b). Forms of Behavior and a New Paradigm of Perception to the Production of New Sounds. In: Roy Ascott (ed), *Art, Technology, Consciousness*. Bristol: Intellectbooks. p. 99 - 103.
- \_\_\_\_\_ (2001). Do Caos à Geração de Novos Timbres. In: XIII ENCONTRO NACIONAL DA ANPPOM, 23 a 27/abril/2001, Belo Horizonte. *Anais do XIII Encontro Nacional da ANPPOM - Música no Século XXI: Tendências, Perspectivas, Paradigmas*. Belo Horizonte: Escola de Música da UFMG, 2001, p. 601-605, v.II.