Multidimensional sound spatialization by means of chaotic dynamical systems

Edmar Soria
Universidad Nacional Autonoma de Mexico, UNAM
Music Technology Graduate Division
Mexico City
kenshi.shinobi@gmail.com

Roberto Morales-Manzanares
Universidad de Guanajuato
Music Department
Guanajuato, Gto.,Mexico
roberto@ugto.mx

ABSTRACT

We present an instrument that explores an algorithmic sound spatialization system developed with the SuperCollider programming language. We consider the implementation of spatial multidimensional panning through the simultaneous use of polygonal shaped horizontal and vertical loudspeaker array. This framework uses chaotic dynamical systems to generate discrete data series from the orbit of any specific system, which in this case is the logistic equation. The orbits will create the path of the general panning structure form vectors of $\mathbb{R}^n$, containing entries from data series of different orbits from a specific dynamical system. Such vectors, called *system vectors* and create ordered paths between those points or system vectors. Finally, interpolating that result with a fixed sample value, we obtain specific and independent multidimensional panning trajectories for each speaker array and for any number of sound sources.

Keywords

NIME, spatialization, dynamical systems, chaos

1. INTRODUCTION

We describe a general framework environmentIn our multidimensional sound spatialization by means of chaotic dynamical systems MSSCDyS . There have been several works which deal with chaotic dynamical systems applied to sound synthesis: Agostino DiScipio, Rick Bidlack, Eduardo Reck Miranda and Nils Peters are just a few examples of authors who have developed research on this topic. However, the field of spatialization has not been as widely explored in that way to seek new means of expression.

We first present, the generalized structure of the main ideas which includes: generating polygonal speakers arrays, chaotic dynamical systems mapping and panning path creation by Splines[1]. We conclude with a presentation of an example in this environment.

[1]https://github.com/crucialfelix/splines

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.
Copyright remains with the author(s).

2. SPEAKER MULTIDIMENSIONAL ARRAY

Symmetrical polygonal objects can be constructed using complex numbers. The images of the n-th roots of unity in the complex plane are the vertices of a regular n-polygon inscribed in the unit circle. Given $z \in \mathbb{C}$ a complex number, and $n \in \mathbb{N}$, the polynomial $z^n = 1$ has exactly n roots and when this roots are plotted in the Argand diagram -the planar representation of the complex numbers- it generates a regular polygon of n sides centered at the origin. The complex numbers ω that satisfy the previous equation are called nth unity roots.

Let $z = r(cos\theta + isin\theta) = re^{i\theta}$ be a complex number in it's polar form. Using De Moivre's formula, we can easily compute such nth roots:

$$(cos\theta + isin\theta)^k = (cosk\theta + isink\theta)$$

Each root can then be plotted to the Argand diagram and form a regular n-polygon inside the complex unit circle. For example, $z^8 = 1$ rises eight roots of the unity:

$$\{1, \frac{\sqrt{2}}{2}+i\frac{\sqrt{2}}{2}, i, -\frac{\sqrt{2}}{2}+i\frac{\sqrt{2}}{2}, -1, -\frac{\sqrt{2}}{2}-i\frac{\sqrt{2}}{2}, -i, \frac{\sqrt{2}}{2}-i\frac{\sqrt{2}}{2}\}$$

which can be visualized as follows:

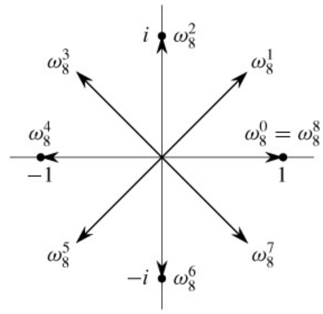

Figure 1: 8th roots of unity.

Using this representation, it is easy to create multidimensional arrays of any numbers of speakers where we situate the listener at the center of the array, i.e at the origin of the polygon inscribed in the circle.

We will start working with a horizontal speaker array considering one listener. The shape of the room can easily be adjusted to be inside this unit circle, moreover, the shape that arises when the listeners are seated in the room can be adjusted for that purposes. For example, if the room has a standard rectangular shape (remember we are only considering now, the horizontal plane), we can shape the listening area to a delimited spot of available seats. In the next figure, the available sitting area is represented by the rhombus which lies inside the rectangle surrounded by an array of six speakers.

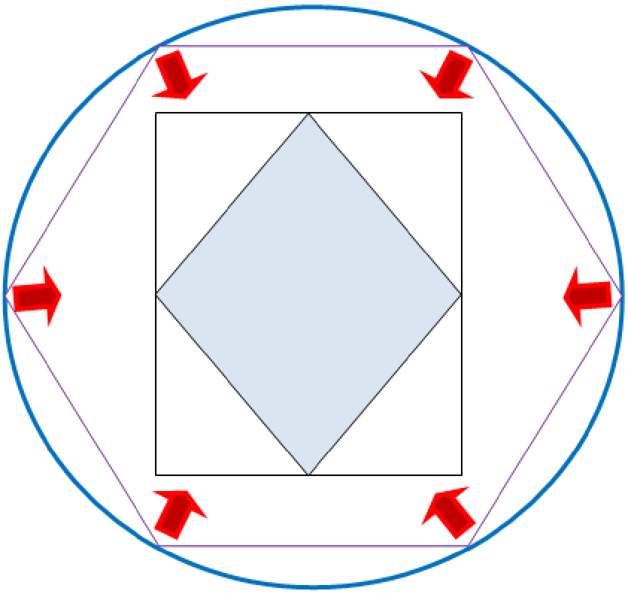

Figure 2: Six-speaker array

We refer to this available seating space for audience as the *listening space* and we encourage experimenting with different configurations for this; circles, ellipses, rectangles, rhombus, triangles, etc, The main reason is that of the reverberation and absorption properties of the audience as a mass mixed with the inherent sonic properties of each hall. If the circular shape of the array does not fit the general shape of the room or the listening space, it can be easily deformed into an ellipse by an anamorphic process for making possible that adjustment.

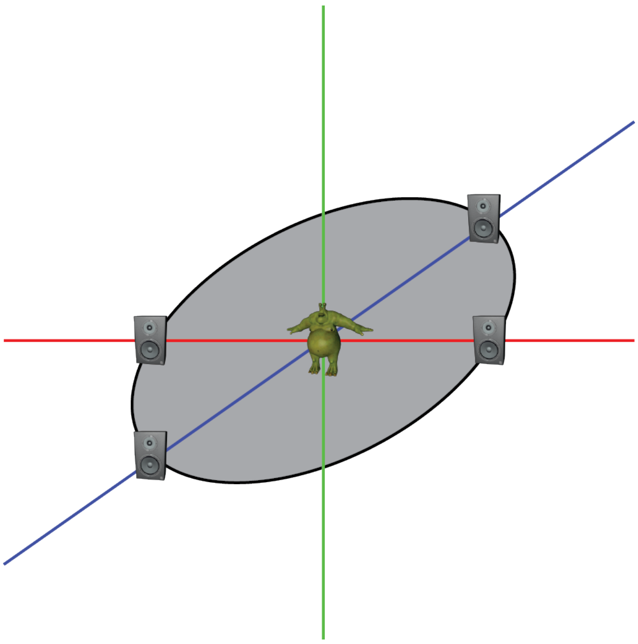

Figure 3: Horizontal speaker array.

This way we can ground the discussion of different configurations of both: the multidimensional speaker array and the shape of the audience inside the room. Further research will need to take into account, of course, psychoacoustic aspects and information that can influence the listener's perception of the final compositional result.

Once the horizontal array is set up, we need to focus on the vertical speaker array. We propose to repeat the last described process, i.e. generate an array of n-speakers placed in the vertex of the polygon generated by the nth complex roots of unity. The difference here lies in the fact that we shall only use the half part of the circumference as an hemisphere array of speakers placed over the listener.

To be more specific, we denote $\mathcal{S}^n = \{s_i : i = \overline{1,n}\}$ the n-dimensional array of speakers consisting of n speakers placed at the vertex of the polygon obtained by finding the nth complex roots of $z^n = 1, z \in \mathbb{C}$.

Since we are working in a 3D environment, the horizontal speaker array will lie in the x-y plane, and we shall refer to it as $\hat{\mathcal{S}}^n_{x,y}$. Now, consider a $\mathcal{S}^n$ placed over the x-z plane, due to the very nature of a regular concert hall we shall only use the positive semicircumference of the z axis; this x-z array will be denoted as $\mathcal{S}^k_{x,z}$, where $k = m/2$ for m to be the desired number of vertices in the complete circumference. For instance, with an array defined by a $\mathcal{S}^5_{x,z}$, and a listener placed exactly at the origin, the listener will perceive one speaker exactly behind -which could be indeed, the same speaker of $\mathcal{S}^n_{x,y}$-, then another one placed at an angle of $\pi/4$ behind, the third one would be exactly up of the listener, the fourth one will be at an angle of $\pi/4$ in front, and the last one would be exactly in front as shown in figure 4.

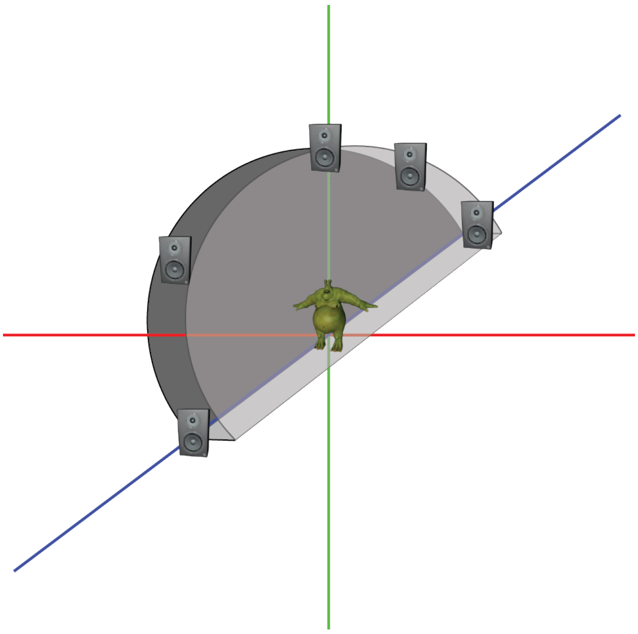

Figure 4: Vertical speaker array

Now we can build our primary and most basic spatial set up of speakers, consisting of a complete circumference in the x-y plane and a semicircumference in the z-x plane as $\mathfrak{S} = \{\mathcal{S}^k_{x,z}, \mathcal{S}^n_{x,y}\}$.

Would it be possible to have others speakers arrays which do not necessarily lie exactly along the x,y or the x,z plane? Given any $\mathcal{S}^n_{x,y}$ we can rotate it for the desired angle $\theta_{x,y}$ respect to the x-y plane and then we translate it up in the z axis until the speaker placed at the lowest position gets the zero high in z - and this process can be done in the inverse order-. For example, we can create a six speaker array $\mathcal{S}_{x,y}^{6}$ from which we would like to get a kind of unusual spatialization placing the left part of the circumference above a certain high in such a way that if a listener is placed at the center, the left speaker will be 5 fts raised over the floor and the right speaker will be exactly at floor level. In this case, we can translate $\mathcal{S}_{x,y}^{6}$ exactly as it is, 5 fts up along the z axis. Then we need to calculate the angle of the rotation which will be $\theta = arcsin\frac{5}{r}$ where r is the radius of the circumference. This allows us to place any array of speakers at any position, orientation or heigh for any purpose and spatial configuration.

We will refer to this more general set up as $\mathcal{S}^{k}(x, y, z, \theta_{x,z}, \theta_{x,y})$ where $\theta_{x,z}, \theta_{x,y}$ are the angles between the speaker array, the xz plane and the xy plane respectively.
Let $\mathcal{S}_{i}^{k}(x, y, z, \theta_{x,z}(i), \theta_{x,y}(i)), i = \overline{1,s}$ be s semicircumferences with two common vertexs among the x axis as is shown in the figures 5 and 6.

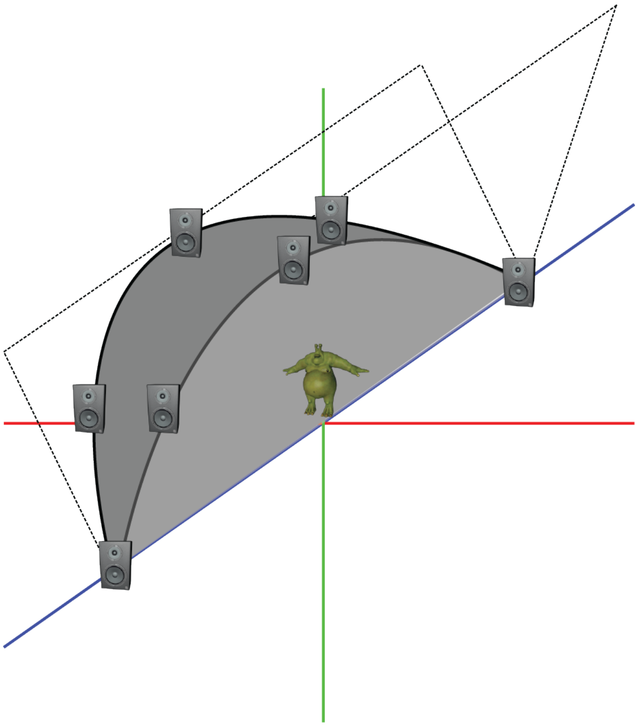

Figure 5: Two-dimensional sonic generating space

If a symmetric distribution is desired, we only have to place every array separated by an angle of $\frac{\pi}{s+1}$ between them inside the x-z plane. This configurations may differ in individual speaker placement for each $\mathcal{S}_{i}^{k}(x, y, z, \theta_{x,z}(i), \theta_{x,y}(i))$ so the sound trajectories appear more complex and interesting the source and perception levels. We refer to this general pre designed configuration of s number of speakers arrays as the *Sonic Generating Space.*

3. ORBIT GENERATING FROM LOGISTIC EQUATION

Recalling the discrete version of the logistic equation:

$$x_n = rx_{n-1}(1 - x_{n-1}) \tag{1}$$

or in the functional form:

$$f(x) = rx(1 - x)$$

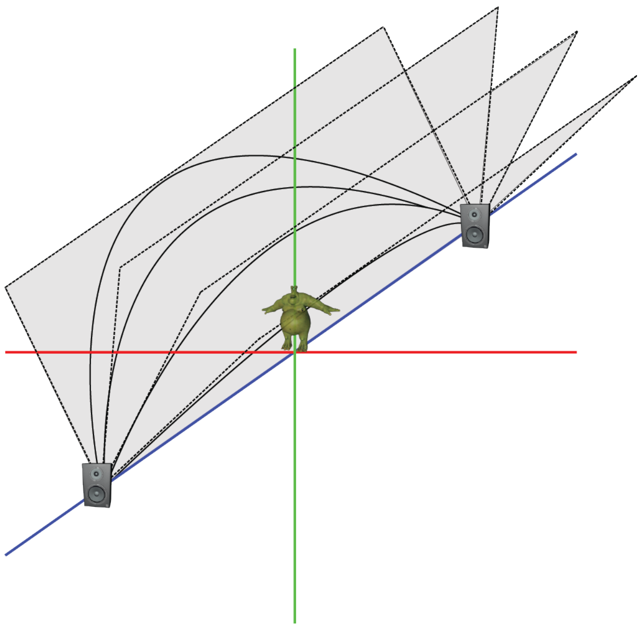

Figure 6: Four-dimensional sonic generating space

This models into a simple dynamical system that presents chaotic behavior under certain conditions.

Defining the *whole orbit* to be the set $\mathcal{O}(x) = \{f^k(x) : -\infty < k < \infty\}$[2], it is immediately seen that $\mathcal{O}(x)$ is a data series that reflects the performance of the equation given some initial conditions on the dependent variable x and the so called control parameter r. We rewrite the elements of the orbit in this fashion:

$$\mathcal{O}(x) = \{x_i : i = \overline{1,n}, x_i = f^i(x)\}$$

As long as the control parameter varies from certain range, it is possible to observe the change in performance of that system, from stable to chaotic. This is shown in the *bifurcation diagram.*

We designed, first an algorithm to evaluate the logistic equation for any set of initial conditions and any suitable range of values. Then we wrote an algorithm for applying the four basic euclidean isometries[3] to the orbits generated by the logistic equation, this allow us to expand our possibilities of getting more orbits with one single evaluation of our dynamical system.

The bifurcation diagram is a very useful tool because we can know in advance which type of behaviour the system will have according to changes in the control parameter. One of the features of our instrument is the freedom to explore a range from static, stable to chaotic trajectories.

[2]This set is the collection of each one of the results of the iteration of f, for example, if $f(x) = x^2$ then $f(f(x)) = f^2(x) = (x^2)^2$, $f(f(f(x))) = f^3(x) = (x^4)^2$, and so on. This generates the so called forward orbit $\mathcal{O}(x) = \{x, x^2, x^4, x^8, .., x^{2^k}, .. : k \geq 0\}$. The term *forward* refers to the fact that only non negative exponents for iterations are considered, i.e, we are tracking the development of the function as it goes forward in time. Sometimes the forward orbit is denoted as $\mathcal{O}^+$, since we will be dealing in this work only with forward orbits, we will use the symbol $\mathcal{O}(x)$ for that general purpose.
[3]This isometries are horizontal and vertical reflection and translation.

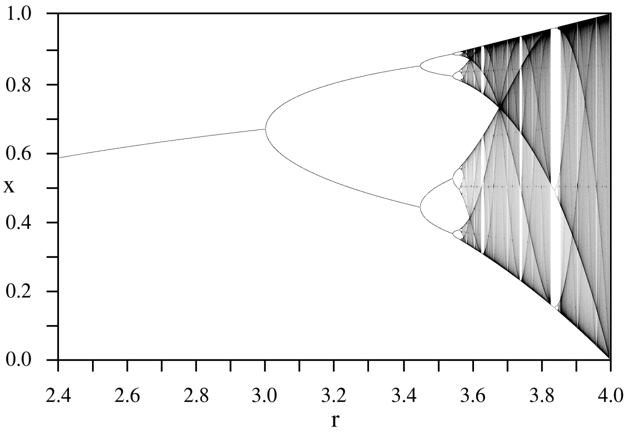

Figure 7: Bifurcation diagram

4. CREATING SMOOTH PATHS

Once we have a specific way to generate orbits from the logistic equation, we define the number s of desired speakers arrays $\mathcal{S}_i^r(x, y, z, \theta_{x,z}(i), \theta_{x,y}(i)), i = \overline{1, s}$ and the configuration of all them as described in the last section. Then we generate an specific orbit for each one of these arrays. However, it should be noticed that at this point we have only considered one single sound source travelling around the listener through the sonic generating space in our general framework. This is of course a reduction of a more general situation. Suppose we have m number of sound sources to be spatializated in our composition. We will have then k_l number of orbits for each sound source, where $l = \overline{1, m}$ and this number may vary from one source to another since we can choose not to use the whole sonic generating space for each one of the sources, i.e. we can choose to use any number of speaker arrays for specific sources. We get the k_l orbits from the logistic equation for each source, then treat them as sequences and finally create a set which will include all of them:

$$\mathbb{O}_j = \{\mathcal{O}_i^j(x) : i = \overline{1, l}\}$$

The last set includes all the orbits of the jth sound source and each orbit is represented as:

$$\mathcal{O}_i^j(x) = \{x_{i_j}\}_{j=1}^n$$

We then define a suitable vector $y \in \mathbb{R}^n$ from elements of $\mathbb{O}_j$. Considering each of these vectors as k dimensional points to be joined with the method of Splines in SuperCollider, we will get a sequence of n points in $\mathbb{R}^k$. The first vector for the jth sound source will be:

$$y_1^j = (x_{1_1}^j, x_{2_1}^j, \ldots, x_{1_k}^j)$$

and so on for the other $n - 1$ vectors. This is how we define the *system vectors*.

Next we connect these points or *system vectors* among themselves using the method of splines for creating smooth paths as transitions between the elements previously described. The interpolated version with a fixed sample of this sequence give us k_l trajectories of the general system for each source. We use these trajectories as actual panning paths among speakers of each array. We can make interpolated trajectories to obtain vertical value range from 0 to 1, so that we divide the interval $[0, 1]$ in exactly $r(i, n)$ pieces where $r(i, n)$ is the number of speakers of the *ith* array of the *nth* source. Assigning horizontal axis to time we get a clear path of the panning of each array of each source through time. In the next example, a simple three dimensional sonic generating space is created by three speakers arrays with different number of speakers each one.

Let $\mathcal{S}_2^3(x, z)$, $\mathcal{S}_2^4(x, y, z, \theta_{x,z} = \frac{\pi}{4})$ and $\mathcal{S}_3^4(x, y, z, \theta_{x,z} = \frac{3\pi}{4})$ be the first,second and third arrays with one,four and four speakers respectively for each one, placed symmetrically at angle of $\frac{\pi}{4}$ each one of the other respect to the x-z plane.

Suppose we have two sources we want to spatialize. We then generate six orbits of 40 iterations from the logistic equation and assign each one of them to each source so that $\mathcal{O}_1^1(x), \mathcal{O}_2^1(x), \mathcal{O}_3^1(x)$ represent the three orbits for the source one and in the same way, $\mathcal{O}_1^2(x), \mathcal{O}_2^2(x), \mathcal{O}_3^2(x)$ are the orbits for the source two. Next we form the system vectors for each source. We begin with source one:

$$y_1^1 = (x_{1_1}^1, x_{2_1}^1, x_{3_1}^1)$$

where $x_{1_1}^1 \in \mathcal{O}_1^1(x)$, $x_{2_1}^1 \in \mathcal{O}_2^1(x)$ and $x_{3_1}^1 \in \mathcal{O}_3^1(x)$. Going on like that we get a sequence of system vectors for the source one:

$$\{y_i^1\}_{i=1}^{40}$$

The process for the second source is exactly the same. We apply the Spline quark from SuperCollider in order to join these points and once interpolation is done, we have three trajectories which can be interpreted as the resulting path from chaotic orbits. Since each trajectory represents each one of the speakers arrays, we divide the interval $[0, 1]$ in to one, tow and three equal segments. For the first array, 0 will represent the first speaker and 1 the second speaker and the trajectory will give us a precise representation in time of the sound movement through this configuration. In the second array, 0 will represent the first speaker,1/4 the second, 1/2 the third and 1 the fourth one, and the trajectory will again represent the sound spatial position over time. This very same method will apply for the third array and for each one of the arrays of the second source. Below we show the spline graphic and the interpolated trajectories in a GUI for this example.

Another useful approach would be to make a planar projection of the whole array system, representing it as a matrix $A \in \mathcal{M}_{\nabla \times \int}$ where r is the max number of speakers of the arrays and s is the total number of arrays. In this way, each entry or individual speaker a_{ij} can be actually considered as a function representing sound amplitude over time: $a_{ij} = (t, f_{ij}(t, s))$ where $(t, f_{ij}(t, s))$ is an ordered pair that reflects the activity of the speaker a_{ij} respect to the sth sound source. In this way we are able to directly map splined chaotic trajectories to an amplitude function over time for each one of the individual speakers. This approach might look more laborious but it gives a more precise control and flexibility when the composer needs some special details about sound trajectories.

5. CONCLUSIONS

This work presents a general framework method for developing a spatialization systems focused on electroacoustic and acousmatic music performance and creation. Although we used the logistic equation as orbit generator, any dynamical system could be explored. Our contribution is intended to be at the very root of the compositional process giving to the creator a tool for exploring new ways for spatial sound placement over time for a wide range of speakers arrangements. The advantage of using controlled chaotic dynamical systems like the logistic equation, lies on the fact that the composer can freely and consciously choose between stable or irregular behaviour for the orbits that will generate his/her panning trajectories. Besides, with the use of isometries, it is possible to generate different related orbits with one single evaluation of the system. The use of the spline method in SuperCollider allows the possibility of joining and relating those values from orbits into a well defined and coherent general system including synthesis parameters in the same way we created panning trajectories.

6. ACKNOWLEDGMENTS

To the Graduate Division of Music of UNAM. To the Universidad de Guanajuato. Special thanks to Roberto Cabezas for creating media images for figures 3,4,5 and 6, as it was a big support for the present work and Adrian Freed for his proofreading and valuable feedback.

7. LINKS

http://ccrma.stanford.edu/.../turenas-_the_realization_of_a_dream-3
http://www.mat.ucsb.edu/res_proj1.php
http://cnmat.berkeley.edu/publication/hci_design_spherical_speaker.
http://www.dolby.com/us/en/professional/technology/cinema/dolby-atmos.html.

8. REFERENCES

[1] Robinson Clark (1999). *Stability, Symbolic Dynamics and Chaos*. CRC Press.

[2] Mocanu Marcelina (2012). *A Problem solving approach to some applications of the roots of unity to regular polygons*. International Journey of Geometry Vol. 1.

[3] Bidlack Rick (2008). *Chaotic Systems a simple (but complex) Compositional Algorithms.* Department of Music, State University of NewYork.

[4] Di Scipio Agostino (2009). *Composition by exploration of Non-Dynamical Systems*. CSC Uiversity of Padova.