Fuzzy Logic Control Tool Kit: Real-time Fuzzy Control for Max/MSP and Pd

Rodrigo F. Cádiz and Gary S. Kendall
Instituto de Música, Pontificia Universidad Católica de Chile
rcadiz@puc.cl
School of Music, Northwestern University
g-kendall@northwestern.edu

Abstract

In computer music, it is often the case that the compositional act and the parametric control of the underlying synthesis algorithms are not separable from each other. In these situations, composition is completely intertwined with and dependent on the control of the synthesis parameters. The compositional control of computer music by fuzzy logic is proposed here. This approach provides a simple, intuitive and flexible control of the compositional process that can be utilized in many different situations. Compositional control is achieved by the fuzzification of the relevant internal synthesis parameters and the parallel computation of several simple fuzzy IF-THEN rules of inference specified by the composer. The proposed approach has been implemented computationally as a package called FLCTK (Fuzzy Logic Control Tool Kit) in the form of external objects for the widely used real-time compositional environments Max/MSP and Pd. As a demonstration of the wide range of situations in which this approach could be used, an example of parametric fuzzy control of granular synthesis is provided.

1 Introduction

When a musical sound is generated computationally, it is important to have a good model with parameters allowing an intuitive manipulation of the sound. The musician-user must be able to develop a musical intuition that will allow experimenting with the synthesis technique used (Serra 1997).

Parameters are an essential and very important part of the musical applicability of a digital synthesis technique. In 1995, Jaffe established ten criteria for evaluating synthesis techniques (Jaffe 1995). Of the ten, four are directly related to the parameters: How intuitive are the parameters? How perceptible are parameter changes? How physical are the parameters? How well behaved are the parameters?

Digital sound synthesis algorithms, even simple ones, typically involve a large number of parameters, which makes them difficult to explore by hand (Dahlstedt 2001). Each synthesis technique has its own set of parameters, and even computational implementations of the same synthesis algorithm could differ in their parametric structure. As numerous and very different synthesis techniques have emerged through time, specific compositional approaches have been established for each one. Instead of using ad-hoc procedures for compositional control, it could be useful to dispose of a more general approach, one that could be successfully applied in a variety of situations and contexts.

The compositional control of computer music by fuzzy logic is proposed as a way of achieving this goal. Fuzzy logic provides a simple, intuitive and flexible control of the compositional process that can be utilized in many different contexts and situations. Compositional control is achieved by a fuzzy inference system, in which user-defined parameters are the inputs and the underlying synthesis parameters are outputs. The transfer function of the fuzzy system is obtained by the parallel computation of several simple fuzzy IF-THEN rules of inference specified by the composer. For a good text of fuzzy systems please refer to (Ross 2004).

The proposed approach has been implemented computationally as a package called `FLCTK` (Fuzzy Logic Control Tool Kit) in the form of external objects for the widely used real-time compositional environments `Max/MSP` and `Pd`.

Section 2 discusses the general idea of composition as parametric control. Section 3 provides a general introduction to fuzzy logic, discussing the advantages of its use in creative endeavors and a review of specific applications of fuzzy logic in music. Section 4 discusses the implementation and features of the fuzzy logic control took kit and section and presents an example of this type of control in the context of granular synthesis.

2 Parametric composition

The idea of parametric control in composition extends far beyond the realms of electronic or computer music. Gerhard and Hepting propose composition as an exploration of parameter space, with each set of musical parameters corresponding to a point in that space (Gerhard and Hepting 2004). The parameters are initially decontextualized, meaning that they can represent any musical or aesthetic quality or process, to the point that the mapping and re-mapping of these parameters to meaningful musical constructs is very much a part of the composition process. Parameter-based composition allows the composer to navigate a large space of musical ideas and pick what is aesthetically desired. Authors have proposed several ways to approach this kind of parametric composition (Dahlstedt 2001) (Gerhard and Hepting 2004). In the case of computer music, one of the most important know-hows that it brings to the act of composition is the fusion of what Xenakis called microcomposition (sound design) with macrocomposition (score design) (Hoffmann 2004). For Stockhausen, any drastic separation between acoustics and music is no longer meaningful where composition includes the synthesis of the sound waves themselves (Stockhausen 1967).

In order to create synthetic sound, parameters of the algorithm need to be adjusted. And when there is no clear demarcation between "composing the sounds" and "composing with sounds", parameter manipulation becomes a compositional act. Some composers explicitly acknowledge this form of composition. For example, when discussing her pieces *Mellipse* and *Dragon of the Nebulla*, Mara Helmuth writes (Helmuth 1996):

> "*Mellipse* is composed of dense sound masses which change gradually due to filtering and other signal processing techniques. *Dragon of the Nebulla* is a granular synthesis composition with timbral gestures created from changing grain parameters".

This kind of composition is related to what Rowe calls *composition by refinement* (Rowe 1993). Interactive music systems contribute to this kind of composition because computer programs react immediately to changes in configuration (parameter control) and input and a user can develop compositional applications by continually refining initial ideas and sketches, up to the development of a complete piece of music.

There is abundant evidence in the literature that composers have employed parameter manipulation of digital synthesis as a compositional technique. For specific examples, please refer to (Cádiz 2006a)

3 Fuzzy logic

Fuzzy logic (Bandemer and Gottwald 1995) (Cox 1994) (Klir and Yuan 1995) (Kosko 1993) (McNeill and Freiberger 1993) is a concept derived from the mathematical branch of fuzzy sets (Zadeh 1965) that applies *multi-valued logic* to sets or groups of objects. Its flexibility, simplicity, and diversity of applicability makes it a very suitable tool for the parametric control of computer music.

In a narrow sense, fuzzy logic refers to a logical system than generalizes traditional two-valued logic for reasoning under uncertainty, allowing multiple values of truth. In a broad sense, it refers to all the theories and technologies that employ fuzzy sets (Yen and Langari 1999). In general, when fuzzy logic is applied to a problem, it is able to emulate aspects of the human reasoning process, quantify imprecise information and make decisions based on vague and incomplete data (Kosko 1993).

Everything is a matter of degree. This statement is known as the *Fuzzy Principle* and it is one of the most important principles in fuzzy logic theory (Kosko 1993). Fuzzy sets have the special property of allowing *partial membership* of its elements. In order words, a given element can only be partially a member of a fuzzy set.

3.1 Advantages of using fuzzy logic

Fuzzy systems provide several advantages for creative applications. Fuzzy systems are powerful and work in a way that resembles some characteristics of human behavior. Parallel computation of fuzzy rules usually reduces the computation time compared to a traditional mathematical approach. Fuzzy systems, due to the fuzzy approximation theorem, enable the approximation of highly non-linear systems with any degree of accuracy. Fuzzy systems are model-free estimators, in consequence, it is not necessary to know any mathematical model in advance to approximate any system. Fuzzy rules can be easily specified in the form of IF-THEN statements, allowing the building of fuzzy systems with simple linguistic terms. Fuzzy logic allows us to build systems using common sense, and the fuzzy rules can be discussed, tuned, and detuned easily.

Fuzzy logic systems have been widely used in a variety of fields, most prominently engineering and control applications (Klir and Yuan 1995) (Kosko 1993), but they have also been applied to other areas as diverse as data analysis (Bandemer and Gottwald 1995), economics, business and finance (Von Altrock 1997), sociology (Dimitrov and Hodge 2002) and geology (Demicco and Klir 2004). Japan is the world wide leader in fuzzy products (Kosko 1997). The most famous Japanese fuzzy application is the control of the subway system in the city of Sendai. Fuzzy logic has been successfully used in products such as cameras, camcorders, washing machines, vacuums, microwave ovens, braking systems and even an unmanned helicopter.

3.2 Fuzzy logic applications in music

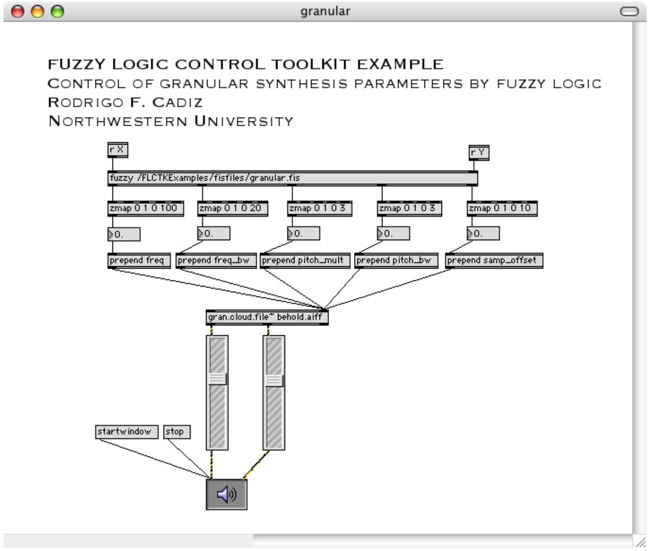
Even though fuzzy logic is mainly an engineering tool, it could be used to solve problems in very different aspects of music and the arts. Examples of the use of fuzzy logic in music are provided below.

Milicevic states that music, unlike language, is a fuzzy concept (Milicevic 1999). In particular, he suggests that listening to computer music represents a fuzzy concept containing high noise capacity or high percentage of *fuzzy entropy*. Assuming that composers of computer music are streaming towards a positive cultural response, Milicevic proposes a fuzzy analysis system that could help them in reducing the high noise capacity when making their musical pieces. Kostek applied soft computing tools including neural networks, fuzzy logic and rough sets to the field of musical acoustics (Kostek 1999). In particular, he designed a fuzzy system to control a pipe organ. Monti and Sandler (Monti and Sandler 2002) developed a blackboard fuzzy inference system that transforms audio into MIDI format. Weyde used a neuro-fuzzy system in the recognition of rhythmic structures (Weyde 2001). Orio and De Piro developed a neuro-fuzzy system for the coding of musical gestures for interactive live performances (Orio and De Piro 1998). Usa and Mochida developed a multi-modal conducting simulator, partially based on fuzzy logic, which behaves like a real orchestra (Usa and Mochida 1998). Kiseliova et at. developed a "fuzzy music interpretation system" (Kiseliova et al. 2005). The general idea is to manipulate a mechanical computer interpretation of a piece of music by applying fuzzy rules coming from an expert performer, so that the outcome corresponds more to a human interpretation. Friberg developed a real-time algorithm for analyzing the emotional expression in music performance and body motion (Friberg 2004). The primary intention is to use the system as a real-time controller in an artistic human-computer performance system. Elsea applied concepts of fuzzy logic to problems in music analysis and composition (Elsea 1995). Lee and Wessel added a real-time interactive fuzzy reasoning system and a neural network simulator to the MAX real-time music programming language (Lee and Wessel 1993). Cádiz and Kendall developed a particle-based sound synthesis algorithm based in fuzzy systems (Cádiz and Kendall 2005, 2006). Miranda and Maia introduced a new model for granular synthesis using Markov chains and fuzzy sets (Miranda and Maia 2005). The Markov chains are used to control the evolution of sound in time and the fuzzy sets are employed to define the internal structure of the sound grains. Cádiz developed a fuzzy logic mapper that maps visual information into aural information and vice-versa (Cádiz 2004, 2006b). A very similar approach was proposed by Schatter, Züger and Nitschke (Schatter et al. 2005).

4 The Fuzzy Logic Control Tool Kit

The FLCTK is built on top libfuzzy, a collection of object-oriented classes and methods developed by the author and written in C++. It is a general purpose library that can be compiled as a static or dynamic library and it is platform-independent. One of the main features of this library is that is has the ability to read and write a fuzzy system description in the fis file format, which is the default format of the widely used MATLAB Fuzzy Logic Toolbox (Mathworks 2006). This means that it is possible to use MATLAB to design a complete fuzzy system, including fuzzy inputs/outputs and rules, save it as a fis file and then use the specified fuzzy model by importing the fis file in to the appropriate FLCTK object.

libfuzzy supports the most common operation and properties of fuzzy sets, and it contains an full implementation of a Mamdani fuzzy inference system as well. For details of fuzzy system theory and operations please refer to (Ross 2004). The FLCTK is basically libfuzzy embedded into Max/MSP and Pd externals. The externals were written using Flext (Grill 2006) in C++.


granular

FUZZY LOGIC CONTROL TOOLKIT EXAMPLE
CONTROL OF GRANULAR SYNTHESIS PARAMETERS BY FUZZY LOGIC
RODRIGO F. CADIZ
NORTHWESTERN UNIVERSITY

Figure 1: Control of granular synthesis with the FLCTK

Figure 1 shows a screenshot of a Max/MSP patch that utilizes the FLCTK's external "fuzzy" to control the synthesis parameter of a granular synthesis algorithm. In this specific case, granular synthesis is done by means of an external called `gran.cloud.file` developed for `Max/MSP` by Nathan Wolek (Wolek 2002). The fuzzy object reads the system specifications from a `fis` file on disk. In this example, the inputs and outputs of the system are shown in table 1. The input variables are user defined and, in this case, they represent two abstract variables, called *X* and *Y*. They could represent anything. In this example, they are controlled by the mouse position in the `Max/MSP` patch. The outputs correspond to a subset of five parameters taken from the parameter space of the `gran.cloud.file` object. The fuzzy system in this case, acts as a "mapper", translating from the two dimensional user defined parameter space to the five-dimensional parametric space that control the sound synthesis. As the user moves the mouse around, the parameters of the synthesis change accordingly to nine fuzzy inference rules specified in the `fis` file and shown below.

Input variables	Output variables
X	Frequency
Y	Frequency Bandwidth
	Pitch multiplicator
	Pitch bandwidth
	Sample offset

Table 1: Input and output variables

The "system" portion of the `fis` file used in this example is provided below:

```
[System]
Name='p1'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=5
NumRules=6
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'
```

These settings tell the fuzzy controller which kind of inference method (implication, aggregation and defuzzification) to use and how AND and OR operators are defined. All input and output fuzzy variables were defined as having three triangular membership functions, with labels "LOW", "MEDIUM", and "HIGH". The rules used in this example are provided in the following matrix format, used by the `fis` file.

```
[Rules]
1 0, 3 2 1 3 2 (1) : 1
2 0, 3 2 0 1 1 (1) : 1
3 0, 1 1 1 3 1 (1) : 1
0 1, 2 3 3 0 3 (1) : 1
0 2, 1 1 1 1 1 (1) : 1
0 3, 1 1 3 2 3 (1) : 1
1 3, 1 2 1 3 2 (1) : 1
2 3, 3 2 2 1 3 (1) : 1
3 3, 3 3 3 3 3 (1) : 1
```

Input and output columns are separated by a comma. The number in each cell corresponds to the index of each fuzzy set of each variable (1, 2, and 3 in this case). A zero means that the rule doesn't care about that particular variable. Between parenthesis is the weight assigned to each rule and separated by a colon is the implication operator to use (1 = AND, 0 = OR). Figure 2 shows the same nine rules in a graphic format. The first two columns (yellow) represent the inputs and the remaining columns (blue) correspond to the outputs. For a given input vector (represented by the two vertical red lines), each rule and its total contribution to the system is calculated. Each rule affects each output in a different proportion. The contributions of each rule to each output are aggregated creating the fuzzy sets shown in the last row. These fuzzy sets are then defuzzified in order to produce the corresponding scalar outputs.

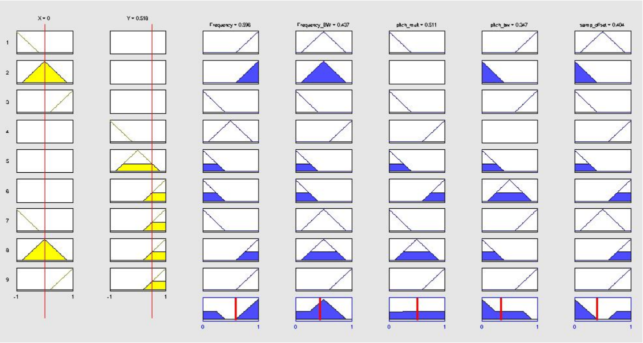


Figure 2: Graphical view of the 9 rules

In summary, the proposed approach allows composers to handle huge and complex parameter spaces by simply specifying IF-THEN rules. In the given granular synthesis example, a five dimensional synthesis parameter space could be successfully explored with only two control parameters.

5 Acknowledgments

This research was partially funded by a Graduate Research Grant from the Graduate School of Northwestern University.

References

Bandemer, H. and S. Gottwald (1995). *Fuzzy sets, fuzzy logic, fuzzy methods with applications*. Chichester ; New York: J. Wiley.

Cádiz, R. (2004). A fuzzy logic model for compositional approaches to audiovisual media. In *Proceedings of the International Computer Music Conference*, Miami, Florida, USA.

Cádiz, R. (2006a, June). *Compositional Control of Computer Music by Fuzzy Logic*. Ph. D. thesis, Northwestern University, Evanston, Illinois, USA.

Cádiz, R. (2006b). A fuzzy-logic mapper for audiovisual media. *Computer Music Journal 30*(1), 67–82.

Cádiz, R. and G. S. Kendall (2005). A particle-based fuzzy logic approach to sound synthesis. In *Proceedings of the Conference on Interdisciplinary Musicology*, Montreal, Canada.

Cádiz, R. and G. S. Kendall (2006). Particle-based sound synthesis: a fuzzy approach. In *SEAMUS National Conference*, Eugene, Oregon, USA.

Cox, E. (1994). *The fuzzy systems handbook : a practitioner's guide to building and maintaining fuzzy systems*. Boston: AP Professional.

Dahlstedt, P. (2001). Creating and exploring huge parameter spaces: Interactive evolution as a tool for sound generation. In *Proceedings of the International Computer Music Conference*, Havana, Cuba.

Demicco, R. and G. J. Klir (2004). *Fuzzy Logic in Geology*. Amsterdam, Boston: Elsevier Academic Press.

Dimitrov, V. and B. Hodge (2002). *Social Fuzziology. Study of Fuzziness of Social Complexity*. Heidelberg,New York: Physica-Verlag.

Elsea, P. (1995). Fuzzy logic and musical decisions. Retrieved online March 2006, from `http://arts.ucsc.edu/EMS/Music/research/FuzzyLogicTutor/FuzzyTut.html`.

Friberg, A. (2004). A fuzzy analyzer of emotional expression in music performance and body motion. In *Proceedings of Music and Music Science*, Stockholm, Sweden.

Gerhard, D. and D. Hepting (2004). Cross-modal parametric composition. In *Proceedings of the International Computer Music Conference*, Miami, Florida, USA.

Grill, T. Flext. C++ programming layer for cross-platform development of PD and Max/MSP externals. Retrieved online March 2006, from `http://www.parasitaere-kapazitaeten.net/ext/flext`.

Helmuth, M. (1996). Multidimensional representation of electroacoustic music. *Journal of New Music Research 25*, 77–103.

Hoffmann, P. (2004). Something rich and strange: Exploring the pitch structure of GENDY3. *Journal of New Music Research 33*(2), 137–144.

Jaffe, D. A. (1995). Ten criteria for evaluating synthesis techniques. *Computer Music Journal 19*(1), 76–87.

Kiseliova, T., H. Kiendl, and Y. Rambinintsoa (2005). Fuzzy rules in computer-assisted music interpretation. In *Proceedings of the International Computer Music Conference*, Barcelona, Spain.

Klir, G. J. and B. Yuan (1995). *Fuzzy sets and fuzzy logic : theory and applications*. Upper Saddle River N J: Prentice Hall PTR.

Kosko, B. (1993). *Fuzzy Thinking. The new science of fuzzy logic*. New York: Hyperion.

Kosko, B. (1997). *Fuzzy Engineering*. Upper Saddle River, New Jersey: Prentice Hall.

Kostek, B. (1999). *Soft Computing in Acoustics. Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics*. Studies in Fuzziness and Soft Computing. Heidelberg, New York: Physica-Verlag.

Lee, M. and D. Wessel (1993). Real-time neuro-fuzzy systems for adaptive control of musical processes. In *Proceedings of the International Computer Music Conference*, Japan.

Mathworks. The fuzzy logic toolbox. Retrieved online March 2006, from `http://www.mathworks.com/products/fuzzylogic`.

McNeill, D. and P. Freiberger (1993). *Fuzzy logic*. New York: Simon & Schuster.

Milicevic, M. (1999). Aesthetics of designing an adaptive fuzzy system for the evaluation of the computer music. In *Proceedings of the International Computer Music Conference*, Beijing, China.

Miranda, E. and A. Maia (2005). Granular synthesis of sounds through markov chains. In *Proceedings of the International Computer Music Conference*, Barcelona, Spain.

Monti, G. and M. Sandler (2002). Automatic polyphonic piano note extraction using fuzzy logic in a blackboard system. In *5th International Conference on Digital Audio Effects*.

Orio, N. and C. De Piro (1998). Controlled refractions: a two-levels coding of musical gestures for interactive live performances. In *Proceedings of the International Computer Music Conference*, Ann Arbor, Michigan, USA.

Ross, T. (2004). *Fuzzy Logic with Engineering Applications*. Chichester: John Wiley & Sons.

Rowe, R. (1993). *Interactive music systems machine listening and composing*. Cambridge, Mass.: MIT Press.

Schatter, G., E. Züger, and C. Nitschke (2005). A synaesthetic approach for a synthesizer interface based on genetic algorithms and fuzzy sets. In *Proceedings of the International Computer Music Conference*, Barcelona, Spain.

Serra, X. (1997). Current perspectives in the digital synthesis of musical sound. *Formats 1*(1).

Stockhausen, K. (1967). The concept of unity in electronic music. *Perspectives of New Music 1*(1), 39–48.

Usa, S. and Y. Mochida (1998). A multi-modal conducting simulator. In *Proceedings of the International Computer Music Conference*, Ann Arbor, Michigan, USA.

Von Altrock, C. (1997). *Fuzzy Logic and Neurofuzzy Applications in Business and Finance*. Upper Saddle River, New York: Prentice-Hall.

Weyde, T. (2001). Grouping, similarity and the recognition of rhythmic structure. In *Proceedings of the International Computer Music Conference*, Havana, Cuba.

Wolek, N. (2002). A Granular Toolkit for Cycling74's Max/MSP. In *SEAMUS 2002 National Conference*, University of Iowa.

Yen, J. and R. Langari (1999). *Fuzzy logic : intelligence, control, and information*. Upper Saddle River, N.J.: Prentice Hall.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control 8*, 338–353.