Fuzzy Logic Control Toolkit 2.0: composing and synthesis by fuzzyfication

Rodrigo F. Cádiz*
Center for Research in Audio Technologies
Music Institute
Department of Electrical Engineering
Pontificia Universidad Católica de Chile
rcadiz@uc.cl

Marie Gonzalez-Inostroza
Center for Research in Audio Technologies
Department of Electrical Engineering
Pontificia Universidad Católica de Chile
mrgonzalez@uc.cl

ABSTRACT

In computer or electroacoustic music, it is often the case that the compositional act and the parametric control of the underlying synthesis algorithms or hardware are not separable from each other. In these situations, composition and control of the synthesis parameters are not easy to distinguish. One possible solution is by means of fuzzy logic. This approach provides a simple, intuitive but powerful control of the compositional process usually in interesting non-linear ways. Compositional control in this context is achieved by the fuzzification of the relevant internal synthesis parameters and the parallel computation of common sense fuzzy rules of inference specified by the composer. This approach has been implemented computationally as a software package entitled FLCTK (Fuzzy Logic Control Tool Kit) in the form of external objects for the widely used real-time compositional environments Max/MSP and Pd. In this article, we present an updated version of this tool. As a demonstration of the wide range of situations in which this approach could be used, we provide two examples of parametric fuzzy control: first, the fuzzy control of a water tank simulation and second a particle-based sound synthesis technique by a fuzzy approach.

Author Keywords

composition, sound synthesis, fuzzy logic

CCS Concepts

•Computing methodologies → Vagueness and fuzzy logic; •Applied computing → Sound and music computing;

1. INTRODUCTION

In computer music, usually the composer manipulates parameters allowing an intuitive generation of the produced sound. Overtime, this develops into a musical intuition that will allow experimenting with the synthesis technique behind the scenes [18].

*Tinker Visiting Professor, Center for Latin American Studies (CLAS) and Center for Computer Research in Music and Acoustics (CCRMA), Stanford University.

NIME'18, June 3-6, 2018, Blacksburg, Virginia, USA.

Parameters are an essential part of the musicality of any digital sound synthesis technique. In 1995, Jaffe established ten criteria for evaluating a particular synthesis techniques [12]. 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?

Even the most simple digital sound synthesis algorithm or hardware synthesizer typically involves a large number of parameters, which makes them difficult to explore by hand [6]. Each synthesis technique has its own and different set of parameters, and even software implementations of the algorithm could greatly differ in their parametric structure. In consequence, specific compositional approaches have been established for each implementation. Instead of using an ad-hoc procedure for compositional control, it seems that a more general approach, one that could be successfully applied in a variety of situations and contexts, would be highly beneficial.

The compositional control of computer music by fuzzy logic has been proposed as a way of achieving this goal [2]. Fuzzy logic provides a simple, intuitive and powerful control of the compositional process that can be situated in many different contexts. Such compositional control is achieved by a fuzzy inference system, in which user-defined parameters are the inputs to the system and the underlying synthesis parameters are outputs. The transfer function of the fuzzy system is obtained by the parallel computation of several simple fuzzy rules of inference specified at will by the composer.

We have previously implemented this approach in a software package called `FLCTK` (Fuzzy Logic Control Tool Kit) [4], in the form of external objects for the widely used real-time compositional environments `Max/MSP` and `Pd`. In this article, we present a new implementation, rewritten from scratch in the Java language, with several new functionalities that we put at the disposal of the NIME community.

This article is structured as follows. 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 new fuzzy logic control took kit and section 5 and presents examples of fuzzy control both in a computer simulation of the water level in a tank and in an particle-based sound synthesis application.

2. PARAMETRIC COMPOSITION

The idea of parametric control in composition is not unique of electronic or computer music. Gerhard and Hepting propose composition as an exploration of a parametric space, where each set of musical parameters corresponding to a point in that space [9]. The parameters initially can represent any musical or aesthetic quality or process, to the point that the mapping of these parameters to meaningful musical constructs is an integral part of the composition process, and composition becomes the navigation of a large space of musical ideas and pick what is aesthetically desired at a given moment. This is particularly true when improvising or performing on stage using complicated machines and synthesizers with a large number of parameters.

Authors have proposed several ways to approach this kind of parametric composition [6] [9]. 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 micro-composition (sound design) with macro-composition (score design) [11]. For some composers such as Stockhausen, any drastic separation between acoustics and music is no longer meaningful where composition includes the synthesis of the sound waves themselves [19]. When creating 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", as Stockhausen clearly states, parameter manipulation becomes a compositional act.

There is abundant evidence in the literature that composers have employed parameter manipulation of digital synthesis as a compositional technique. For specific examples, we refer the reader to [2].

3. FUZZY LOGIC

Fuzzy logic [1] [5] [13] [14] [16] is a concept derived from the mathematical branch of fuzzy sets [22] that applies *multi-valued logic* to sets or groups of objects. Its flexibility, simplicity, and diversity of applicability makes it a 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 [21]. 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 [14].

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 [14]. 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 [13] [14], but they have also been applied to other areas as diverse as data analysis[1], economics, business and finance [20], sociology[8] and geology [7]. Fuzzy logic has been successfully used in products such as cameras, camcorders, washing machines, vacuums, microwave ovens, braking systems and unmanned vehicles.

4. THE FUZZY LOGIC CONTROL TOOL KIT 2.0

The `FLCTK` version 2.0, is an improved version over the original `FLCTK`, which was built on top `libfuzzy`, a collection of object-oriented classes and methods developed by the author and written in `C++` [4]. In contrast, version 2.0 is built in Java, to be integrated with the `mxj` engine of MaxMSP[1], which allows to run Java-based externals inside the Max environment, and has also been ported to Pd[2]. One of the main features of the `FLCTK` 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[3]. 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. Version 2.0 also has the capability of doing is in the opposite direction: designing a fuzzy system in MaxMSP or Pd and then open it in MATLAB for posterior use.

`FLCTK` 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 [17].

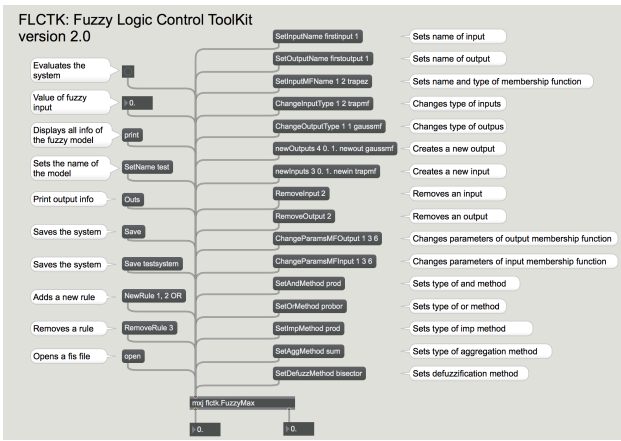

Figure 1: Main help file of the `FLCTK` in the Max7 environment.

Figure 1 shows a screen-shot of a `Max7` patch that utilizes the `FLCTK`'s external "Fuzzy" with all its options. The fuzzy object reads the system specifications from a `fis` file on disk or they can be created by sending messages to the Fuzzy

[1]http://cycling74.com
[2]https://puredata.info/downloads/pdj
[3]https://www.mathworks.com/help/fuzzy/index.html

object. In this example, details of the fuzzy system in use are displayed in the Max window.

There are messages that 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 can be defined defined as having triangular or Gaussian membership functions, with labels specified by the user.

4.1 Download

The FLCTK can be downloaded from its github repository at `https://github.com/rcadiz/flctk.git`. The package contains the source Java code, compiled code, help files and examples, some of which are detailed below.

5. EXAMPLES

5.1 Water level control in a tank

First, we present a computer simulation of the control in a water tank. We chose this example simply because it is also included in MATLAB's fuzzy logic toolbox[4]. This provides a way to compare the performance and the correct functioning of our toolbox in relation to the one that MATLAB provides.

For this system, the water that flows into the tank is controlled using a valve. The outflow rate depends on the diameter of the output pipe, which we assume to be a constant, and the pressure in the tank, which depends on water level. Therefore, the system has nonlinear characteristics and is ideal to be handled by a fuzzy control strategy.

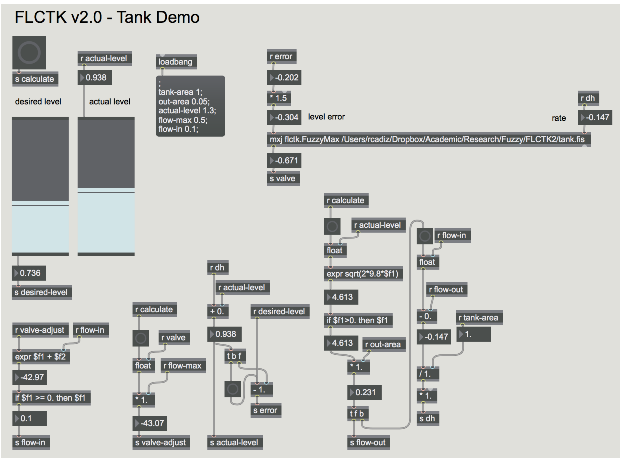

Figure 2: Automatic water level control of a tank simulation, implemented in Max7 using the flctk.

5.2 A particle-based sound synthesis algorithm based on fuzzy logic

An audio synthesis technique based on sound particles, behavioral rules and fuzzy logic is presented as a second example. This technique has been shown to allow for the creation of highly complex sonic behavior by very simple means [3]. Each sound particle in the system possesses several abstract fuzzy properties, labeled as frequency, intensity, spatial position and charge. These properties are the inputs to a rule-based fuzzy logic inference system that controls the temporal evolution of the particles and the interactions between them. The rules are easy to elaborate, modify and adjust with the FLCTK, leading to the creation of very complex sonic phenomena. This is a very good example when composition becomes the manipulation of sound synthesis parameters.

All the previously mentioned properties are fuzzy variables, which means that each one of them consists on several fuzzy sets or membership functions, and values of the variable can have partial membership to them. Table 1 displays all the inputs and outputs used in the proposed fuzzy inference system. Note that time is also included as an input, allowing the possibility to generate time-dependent behaviors or trajectories.

Input variables	Output variables
Frequency	Frequency
Charge	Charge
Intensity	Intensity
X	X
Y	Y
Z	Z
Ext. influence	
Time	

Table 1: Input and output variables for a particle-based sound synthesis algorithm. Each sound particle in the system possesses several abstract fuzzy properties, such as frequency, intensity, spatial position and charge. The fuzzy inference system converts combination of the input parameters into outputs, which are subsequently mapped into sound parameters.

As the fuzzy variables *frequency*, *charge*, *intensity*, *X position*, *Y position*, and *Z position* are also used outputs to the system, we should clarify that this means that the system calculates the new properties at any given time step, based on the values of the previous ones, the influence of external factors and the appropriate time instant.

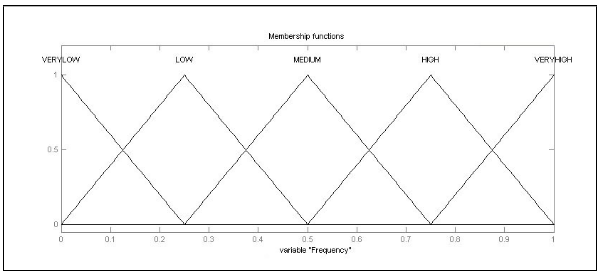

Figure 3: Membership functions for the fuzzification of the variable frequency.

Figure 3 shows as an example the fuzzyfication of the variable *frequency*. In this case, the fuzzy variable consists on five membership functions labeled "VERY LOW", "LOW", "MEDIUM", "HIGH" and "VERY HIGH". Each value of the variable *frequency* belongs in a particular degree to each one of the five membership functions. For example, a *frequency* of 0.5 belongs a 100% to the "MEDIUM" region and a *frequency* of 0.6 belongs approximately 40% to the "MEDIUM" region and 60% to the "HIGH" region.

All the other variables were defined in terms of similar membership functions. In the case of the spatial fuzzy variable, the fuzzy sets were labeled "FAR-" (far to the left), "CLOSE-" (close to the left), "VERY CLOSE", "CLOSE+" (close to the right), and "FAR+" (far to the right).

Some examples of the fuzzy rules that can be used are:

[4] `https://www.mathworks.com/help/fuzzy/examples/water-level-control-in-a-tank.html`

- If frequency is HIGH and time is MEDIUM then frequency is VERY HIGH
- If intensity is VERY LOW then frequency is LOW, charge is HIGH, and X, Y and z are far to the left
- If charge is LOW then charge is HIGH
- If frequency is VERY HIGH then X, Y and z are CLOSE to the right
- If charge is HIGH and time is SHORT then frequency is VERY HIGH
- If external influence is HIGH then charge is LOW and intensity is LOW
- If charge is HIGH and z is far to the right then frequency is LOW and charge is MEDIUM

It should be clear to the reader by now that this process of creating fuzzy variables and rules is very simple and based on common sense if the goal is to control, for example, a physical system, and it can be done in any way imaginable when the result is to be heard as music! In this sense, the design of the fuzzy systems is equivalent to composing the music.

The trajectories of each particle are then determined by these fuzzy inference rules. Each particle can be mapped, for example to a sinusoidal wavetable digital oscillator. All the particles had the same initial frequencies, intensities and spatial positions, but random initial charges. The frequency and intensity variables were used directly as the parameters of each oscillator. The spatial and charge variables were used in the calculation of the external influence of the system on each particle, which in turn modified the resulting subsequent frequencies and intensities.

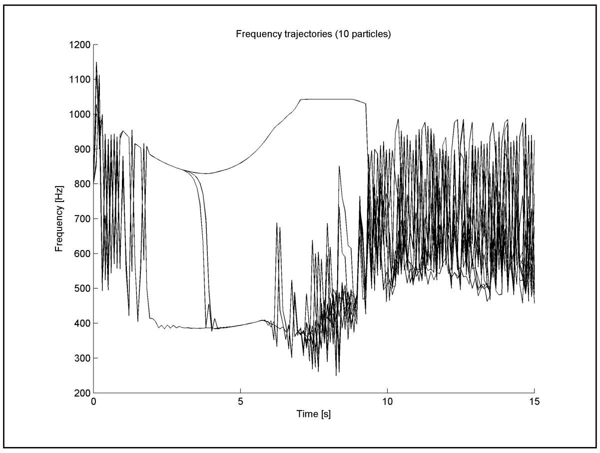

Figure 4: Frequency trajectories in time for a 10-particle system. As it can be seen, highly complex behavior can be generated with a few simple if-then rules.

Figures 4, and 5 show the frequency and intensity trajectories followed by an instance of 10 particles. It is possible to appreciate that the trajectories are quite complex, and in general the particles behave radically different in relation to time. Sometimes they behave very chaotically and some other times, in this example most notably between seconds 1 and 6, they follow smooth and well-defined trajectories. They appear also to be clustered into two different groups.

Overall, the different trajectories shown in the above figures are very complex and possibly chaotic. This is consequence of the power of fuzzy logic: with simple linguistic if-then like rules and the proper fuzzyfication of the parameters of interest, with no knowledge whatsoever of the mathematical nature of the underlying dynamical systems, it is possible to generate very complex behavior.

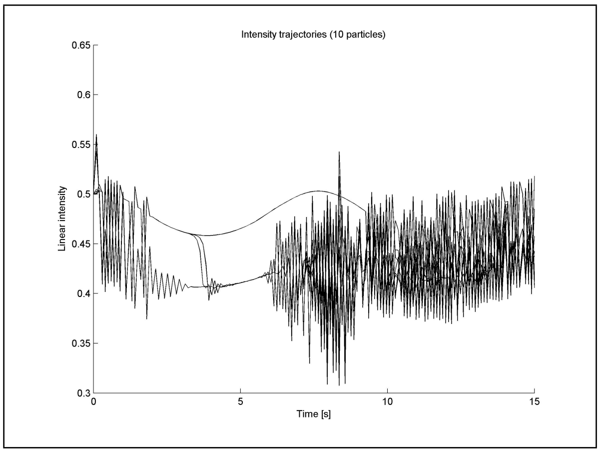

Figure 5: Intensity trajectories in time for a 10-particle system. As it can be seen, highly complex behavior can be generated with a few simple if-then rules.

6. DISCUSSION AND FUTURE WORK

Fuzzy logic is a powerful way to implement non-linear mappings and intuitive control of a high number of non-intuitive synthesis parameters. However, one of the weaknesses of a fuzzy logic approach to parametric composition would be the time required to appropriately design adequate rules for the inference system. In engineering control applications, these rules are derived from expert knowledge or machine learning processes, where the rules are derived from trained data. In artistic applications, these rules constitute the heart of the underlying parameter mapping and it becomes really hard to select appropriate rules for a specific desired output when the parameter space is highly dimensional, which is often the case. Rule specification becomes and art form in itself, and it requires time and the development of an expert knowledge specific to this kind of composition.

In terms of future work, we are currently implementing other applications of this toolbox. In particular, we are designing a system that would capture hand gestures in a multi-touch 3D interface and produce appropriate sound synthesis parameters. We believe that fuzzy logic would be an ideal tool for gesture to sound mappings. Another line of work is the implementation of a real-time fuzzy-based audio equalizer based on musical genre. A non real-time version of this work has been reported in [10].

7. CONCLUSIONS

In summary, the proposed `FLCTK` allows composers and electronic musicians to handle complex parameter spaces by simply defining a simple fuzzy logic system and specifying inference rules that relate input to output fuzzy variables. For example, in the given particle-based synthesis method, the proposed approach is able to generate very complex sonic phenomena by simple means.

Overall, the `FLCTK` is a simple way of designing and implementing fuzzy logic inference systems inside Max or PD. It's compatibility with MATLAB's fuzzy logic toolbox also allow to use this environment in the design and test stages of the fuzzy models.

8. ACKNOWLEDGMENTS

This research was partially funded by Fondecyt Grant #1161328, Government of Chile, and a professorship from the Tinker Foundation and the Center for Latin American Studies, Stanford University.

9. REFERENCES

[1] H. Bandemer and S. Gottwald. *Fuzzy sets, fuzzy logic, fuzzy methods with applications.* J. Wiley, Chichester ; New York, 1995.

[2] R. Cádiz. *Compositional Control of Computer Music by Fuzzy Logic.* PhD thesis, Northwestern University, Evanston, Illinois, USA, June 2006.

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

[4] R. F. Cádiz and G. S. Kendall. Fuzzy logic control tool kit: Real-time fuzzy control for max/msp and pd. In *Proceedings of the International Computer Music Conference*, 2006.

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

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

[7] R. Demicco and G. J. Klir. *Fuzzy Logic in Geology.* Elsevier Academic Press, Amsterdam, Boston, 2004.

[8] V. Dimitrov and B. Hodge. *Social Fuzziology. Study of Fuzziness of Social Complexity.* Physica-Verlag, Heidelberg,New York, 2002.

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

[10] M. Gonzalez, P. de la Cuadra, and R. F. Cádiz. Fuzzy equalization of musical genres. In *Proceedings of the International Computer Music Conference*, 2015.

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

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

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

[14] B. Kosko. *Fuzzy Thinking. The new science of fuzzy logic.* Hyperion, New York, 1993.

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

[16] D. McNeill and P. Freiberger. *Fuzzy logic.* Simon & Schuster, New York, 1993.

[17] T. Ross. *Fuzzy Logic with Engineering Applications.* John Wiley & Sons, Chichester, 2004.

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

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

[20] C. Von Altrock. *Fuzzy Logic and Neurofuzzy Applications in Business and Finance.* Prentice-Hall, Upper Saddle River, New York, 1997.

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

[22] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.