INTRODUCING FUZZY LOGIC AND COMPUTING WITH WORDS PARADIGMS IN REALTIME PROCESSES FOR PERFORMANCE ARTS

Alain Bonardi
IRCAM-CNRS – IMTR Team
alain.bonardi@ircam.fr

Isis Truck
LIASD – EA 4383
truck@ai.univ-paris8.fr

ABSTRACT

When trying to introduce qualitative information into processes for performance arts, two major points must be considered: how to *really* deal with qualitative data and how to insert this into the processes.
We propose to design a library for Max/MSP with the aim of offering a wide and full range of tools to be able to reason with all kinds of qualitative concepts. The idea is to implement inside Max/MSP the well-known fuzzy logic concepts, considering the numerous ways to represent imprecise data.

1. INTRODUCTION

Fuzzy logic is often confused with fuzzy control. People have heard of autonomous control of such devices as boilers, cars or watering systems, and this is often considered as the alpha and omega of fuzzy logic. Even in the academic field, some open source fuzzy logic libraries in fact deal with fuzzy control. It is for instance the case of *jFuzzyLogic* [13] or *Free Fuzzy Logic Library* [11]. Between fuzzy logic and fuzzy control, there are of course many common points but also differences: fuzzy control uses fuzzy logic concepts, that are specialized to optimize the computerized implementation [6]. In fuzzy control, observed facts are precise, represented as singletons and therefore the generalized modus ponens is reduced to fuzzy implication; moreover fuzzy implications are based on AND conjunction: $A \Rightarrow B$ is translated into *A AND B*, which is implemented either by a *min* (Mamdani implication) or *product* operation (Larsen implication). This approach does not enable for instance to deal with the equivalence in classical logic between $A \Rightarrow B$ and $\overline{B} \Rightarrow \overline{A}$, which requires other fuzzy implications, for instance Reichenbach, Willmott or Rescher-Gaines ones.

In the field of real-time interactive music, fuzzy logic has nearly been absent from research and development efforts. A few authors have proposed Max/MSP patches to enable the use of some AI techniques. As an example, in a recent work [9] Eigenfeldt describes methods to create networked multi-agents within Max/MSP and also some fuzzy logic ratings for a drum ensemble. Less recently Elsea [10] explains how the core concepts of fuzzy logic and fuzzy control may be applied to problems of musical decisions, for instance chord inversions.

Computing with Words (CW) [16] paradigm is a methodology in which the objects of computation are words and propositions drawn from a natural language; it is inspired by the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. CW has shown to fit very well all kinds of human expressions (feelings, perceptions, emotions, impressions, believes, etc.). We are interested in introducing these concepts in the world of real-time processes for performing arts. Our research is based on two concurrent tasks: exploring the necessary concepts in fuzzy logic being not limited to fuzzy control, that is relevant high level representations of phenomena that are not only useful as approximation tools for control; identifying case studies showing the relevance of fuzzy approaches and concepts. This double approach led us to specify and develop our own fuzzy library adapted to real-time formalisms and opened beyond fuzzy control. It is named *FuzzyLib*. At the same time we started using it for precise applications: first, for the semantic control of effects in a theatre play; secondly, for fuzzy filtering and decision in gesture recognition.

In this paper, we will first of all recall the design of our fuzzy logic library for Max/MSP before explaining the novelties to overcome the simple fuzzy control. We then present two case studies and evoke future work.

2. FUZZYLIB, A FUZZY LOGIC LIBRARY FOR MAX/MSP

We have developed a software library named *FuzzyLib* for the realtime environment Max/MSP [3]. This library is currently available on Internet at http://imtr.ircam.fr. We have designed three objects that correspond to three levels of fuzzy logic: fuzzification, reasoning and defuzzification, fuzzy rule expression. We present these three objects and then give the main features of the library.

2.1. Linguistic variable handling for fuzzification

The first step in fuzzy logic is fuzzification. The purpose is the semantic representation of a phenomenon usually handled with digital values. Thanks to a range of values provided by an expert and to a certain (generally odd) number of fuzzy subsets, one can represent this phenomenon as a linguistic variable.

We have designed an object named *lv*, which stands for 'linguistic variable'. A screenshot of its help patch is given on figure 1, and shows the fuzzification of the quantity of movement of a performer using five fuzzy subsets: 'still', 'nearlyStill', 'slightlyMoving', 'moving', 'veryMoving'. These labels are provided by the user thanks to the *outFormat* message entering the left input of the object. The right input collects the numerical boundaries (in our case from 0 to 127) of the variable as supposed to be known by an expert. On the right frame of the patch, one can see the various parameters of the object, mainly the shape of fuzzy subsets (triangular or trapezoidal) and the type of partitioning. We have implemented two partitioning possibilities: either uniform or based on an algorithm already published for automatic partitioning according to training data [4] [5]. This option is interesting to quickly set a linguistic variable from a set of training data.

A module for linguistic variable and fuzzy subset graphical display has also been developed. It receives data from the 'lv' object and displays fuzzy subsets and the current value of the variable.

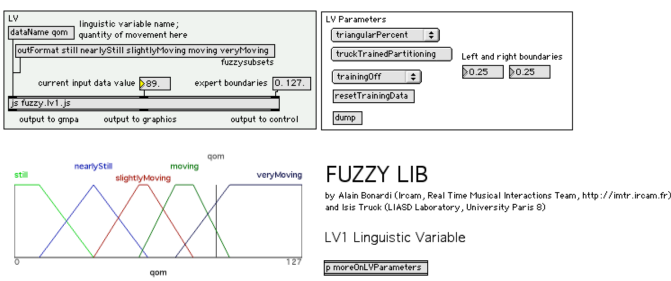

Figure 1. A screenshot of the help patch of the 'lv' object (Linguistic Variable).

2.2. Reasoning and defuzzification level

Reasoning is computed thanks to the generalized modus ponens that enables to solve the following syllogism: if we assume $A \Rightarrow B$ and we get A' close to A as input, what should the answer B' be?

We have developed an object named *gmpa* for Max/MSP, which stands for 'generalized modus ponens application'. A screenshot of its help patch is given on figure 2. The 'gmpa' object receives both:

- data from 'lv' objects: declaration of new linguistic variables, removal of previous variables.
- fuzzy rules connecting input linguistic variables with output linguistic variables. The rules are expressed as Max/MSP messages.

The object also provides defuzzification (by barycentre or maximum methods) for linguistic variables implied as consequents of fuzzy rules. The user has to choose the fuzzy implication and the t-Norm and t-CoNorm to be handled (on the screenshot, *Reichenbach* fuzzy implication has been selected, whereas *Min()* and *Max()* have been chosen as t-Norm and t-CoNorm.

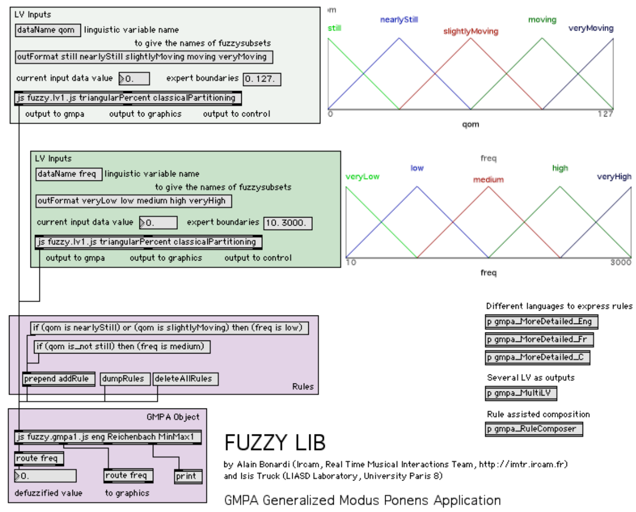

Figure 2. A screenshot of the help patch of the 'gmpa' object (Generalized Modus Ponens Application) connecting two linguistic variables (qom and freq) and taking into account fuzzy rules.

2.3. An assistant for fuzzy rule composing

We also have implemented an interface object named *ruleComposer*, associated with the 'gmpa' object to help users to write fuzzy rules and avoid syntax errors. This object automatically collects the names of the linguistic variables declared with 'lv' objects and provides help to write fuzzy rules. Figure 3 gives a screenshot of its patch.

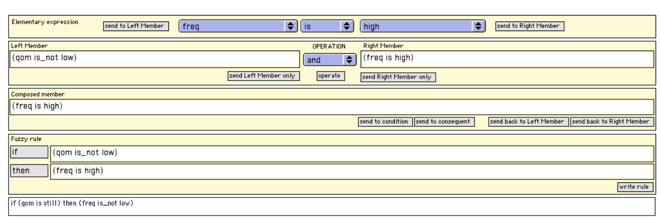

Figure 3. A screenshot of the help patch of the 'ruleComposer' object.

2.4. Main features of *FuzzyLib*

FuzzyLib provides realtime fuzzy logic handling. As we said in the introduction of the paper, our purpose is to design a framework that fits Max/MSP paradigm and also is not limited to fuzzy control. In that purpose, we include the computation of all classical fuzzy implications proposed by literature. Classical t-Norms and t-CoNorms are also implemented.

For its first version, we have made a simplification to quickly sketch the environment. We have considered all measures provided by captors as precise ones. In this case, the General Modus Ponens computation is limited to the fuzzy implication involved, and that enables less complex algorithms.

It means this first version of *FuzzyLib* overcomes fuzzy control on the implication side, but it does not implement yet the full fuzzy logic paradigm with imprecise measures as input. This will be implemented in version 2 after studying the optimization of computation of the generalized modus ponens [8] [15].

3. TWO CASE STUDIES OF APPLICATION

3.1. Semantic control of a crowd generator

To put in practice our library, we have proposed to M. Bataille-Testu, the stage director of a theatre play called *Les petites absences*, to use it during a show, with the imperative constraint of realtime.

He used Max/MSP enhanced with our *FuzzyLib* to control a crowd sound generator that would fit the behavior of the main actor from a video and voice capture and analysis. As a non-specialist, he could very easily define rules to be activated depending on the cases he wanted to point out. Two levels of fuzzy rules were used: first one to identify voice and motion particular states; the second one to map these states to the control of the audio crowd generator. The result is very encouraging since the stage director was able to overcome his difficulties to develop directly in Max/MSP; and the play has continuously been performed since december 2008 with a satisfying (especially for the artistic team) fuzzy-driven interaction.

3.2. Fuzzy decision and filtering in gesture recognition and following

We have outlined another application of the *FuzzyLib* that consists in coupling HMM-based gesture recognition and following with fuzzy filtering and decision. We have experimented character recognition in handwriting starting from previous work at IRCAM by Bevilacqua & al. [2] implementing in Max/MSP HMM gesture recognition and following: the decoding of the result following the various likelihood curves is neither easy nor convenient. Generally users take into account the result given at the very end of the process. We have added a fuzzy level that receives instantaneous information about the likeliest gesture and its likelihood during the gesture presented. It enables to filter with fuzzy decisions non-significant fragments of gestures, and to provide a synthetic semantic abstract of the gesture.

For instance, we have learnt three characters in the system: ‘c’, ‘o’ and ‘b’. Figure 4 shows the abstract generated when drawing a ‘c’: “At the beginning of the movement the likeliest was the first half of gesture #1” (gesture # 1 is ‘c’). It is a kind of semantic overview of the gesture. Figure 5 shows the abstract generated when drawing an ‘o’, which is ambiguous since the beginning is the same as with the ‘c’: “At one half of the movement, the likeliest was the third quarter of gesture #1 (‘c’). At four fifths of the movement, the likeliest was the last quarter of gesture #2 (‘o’)”.

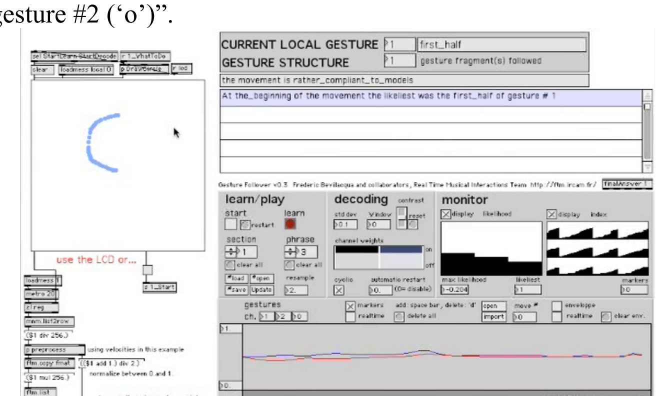

Figure 4. A screenshot of the gesture recognition and following patch, in the situation of handwriting a ‘c’.

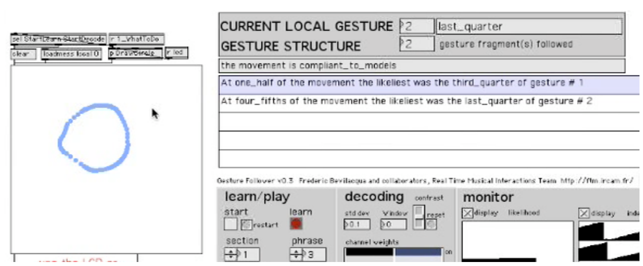

Figure 5. A screenshot of the gesture recognition and following patch, in the situation of handwriting an ‘o’.

We can easily imagine applications of this kind of systems to interactive device that would assist dancers or musicians in learning precise gestures. Next step of this research will consist not only in following the likeliest gesture but the various features of all the gesture models when presenting a new gesture.

3.3. Necessity of a full implementation of fuzzy logic paradigms

For simple mapping between input and output values, a simple fuzzy control is sufficient, and *FuzzyLib* is perfectly suitable for that.

There are however applications that would require a full implementation of fuzzy logic paradigms especially of the generalized modus ponens:

- Situations handling imprecise values as inputs. This can be due to many reasons: bad quality of sensors, low frequency of measures, etc.
- Imprecise reasoning: as soon as we handle an intermediary level of variables between input and output ones, we need to handle fuzzy subsets and not only precise values. For instance, in *Les petites absences*, we build an intermediary variable named *performerGlobalPosition* from such input variables as the bounding box stretching, etc. At present time, since we cannot handle imprecise values we need to fuzzify and defuzzify all values coming in and out variables, passing only numerical values to other variables. We should be able with the version 2 of *FuzzyLib* to pass fuzzy subsets between modules.
- Situations handling particular values as inputs, especially input values coming directly from humans (i.e. expressed through linguistic assessments). In these situations, only *words* are available, no fuzzy sets. Two different formalisms has been proposed, one in [12] and the other in [11], to handle words directly without the help of an expert: words are proposed on a scale (with balanced or unbalanced term sets) and the whole process for the reasoning seems to be transparent, giving finally an outcome as a word or possibly a fuzzy number. However only aggregation and comparison tools have been studied and proposed, so one has to look for the best combination between those tools to propose a convenient reasoning for our framework.

4. CONCLUSION

In this paper we have recalled the spirit of our work since the very first version of our fuzzy library *FuzzyLib*. The current version goes further than the simple fuzzy control in proposing all the implication operators that exist in the literature. We have seen that in practice the *FuzzyLib* is very useful and corresponds to a real need. That is why we will continue this work by proposing an end-to-end linguistic treatment of the fuzzy inference, including all kinds of input values; an approach that could better match user's needs and intentions.

5. REFERENCES

[1] Akdag, H., Truck, I., "Uncertainty Operators in Many-valued Logic", chapter in Encyclopedia of Data Warehousing and Mining - 2nd Edition, John Wang, ed., Information Science Reference, 2008.

[2] Bevilacqua, F., Guédy, F., Fléty, E., Leroy, N., Schnell, N., "Wireless sensor interface and gesture-follower for music pedagogy", in *Proceedings of the International Conference on New Interfaces for Musical Expression* (NIME'07), pp. 124-129, New York, 2007.

[3] Bonardi, A., Truck, I., "Designing a library for computing [performances] with words", *Intelligent Decision Making Systems*, Proceedings of the 4th International ISKE Conference, Hasselt: World Scientific, pages 40-45, 2009.

[4] Bonardi, A, Truck, I., Akdag H., "Building Fuzzy Rules in an Emotion Detector", in *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (IPMU'2006), pp 540 - 546, Paris, 2006.

[5] Bonardi, A., Truck, I., "First steps towards a digital assistant for performers and stage directors", in *Proceedings of the 3rd International Conference on Sound and Music Computing* (SMC'06), pp 91-96, Marseille, 2006.

[6] Bouchon-Meunier, B., "La logique floue et ses applications", Addison-Wesley, 1996.

[7] Camurri, A., Catorcini, A., Innocenti, C., Massari, A., *Music and multimedia knowledge and reasoning: the harp system*, Computer Music Journal, 19(2):34-58, 1995.

[8] Dubois, D., Martin-Clouaire, R., Prade, H., "Practical computing in fuzzy logic", in *Fuzzy Computing*, M.M. Gupta and T. Yamakawa (Editors), Elsevier Science Publishers B.V. (North-Holland), 1988.

[9] Eigenfeldt, A., "Drum Circle: Intelligent Agents in Max/MSP", in *Proceedings of the International Computer Music Conference*, 2007.

[10] Elsea, P., *Fuzzy Logic and Musical Decisions*, University of California, Santa Cruz, 1995. http://arts.ucsc.edu/EMS/Music/research/FuzzyLogicTutor/FuzzyTut.html.

[11] FFLL. Free fuzzy logic library. http://ffll.sourceforge.net/index.html.

[12] Herrera, F., Martinez, L., "A 2-tuple fuzzy linguistic representation model for computing with words", in *IEEE Trans. Fuzzy Systems*, 8(6):746-752, 2000.

[13] jFuzzyLogic. Open source fuzzy logic library and fcl language implementation. http://jfuzzylogic.sourceforge.net/html/index.html.

[14] Kaufmann, A., *Introduction to the Theory of Fuzzy Subsets*, New York, San Francisco, London, 1975.

[15] Martin-Clouaire, R., *Semantics and Computation of the Generalized Modus Ponens: The Long Paper*, International Journal of Approximate Reasoning 3:195-217, 1989.

[16] Zadeh, L.A., "From computing with numbers to computing with words: From manipulation of measurements to manipulation of perceptions", in *International Journal of Applied Math and Computer Science*, 12(3):307-324, 2002.