Fuzzy Interpretation of Music

Harro Kiendl[1], Tatiana Kiseliova[1, 2, 3], Yves Rambinintsoa[1]

[1]University of Dortmund
Department of Electrical Engineering
Chair of Electrical Control Engineering
D-44221 Dortmund
[2]Univesity f Dortmund
Department of Computer Science, Chair I
D-44221 Dortmund
[3]Ivane Javakhishvili Tbilisi State University
Department of Applied Mathematics and Computer Science
380028 Tbilisi, Georgia

kiendl@hyperfuzzy.de
kiseliova@ls1.cs.uni-dortmund.de
su0813@studserver.uni-dortmund.de

Abstract

The modeling of the behavior of human process operators using fuzzy methods is well established in technical applications. We investigate whether these fuzzy strategies can also be successfully applied for modeling the manner in which a musician interprets a piece of music. For L. v. Beethoven's composition "Für Elise" we set up a base of 150 fuzzy rules that relate to features laid down explicitly or implicitly in the score, and produce situation-dependent variations of the volume and time points of the start and end of the notes. For sensitive processing of these rules advanced fuzzy-strategies are applied. The interpretation obtained is encouraging. We are aware that human artistry in interpreting music is much more sophisticated than what we can presently imitate. However, we consider it as a challenge, not an *a priori* fruitless question, to determine to what extent it is possible to automate the interpretation of music by fuzzy methods. At the same time, we consider the problem, due to its complexity, as being an ideal test bed for evaluating existing and developing advanced fuzzy methods.

Keywords: fuzzy systems, knowledge-based fuzzy, data-based fuzzy, modeling, TOR defuzzification, positive and negative rules, interpretation of music.

1 Introduction

For thousands of years music has been an essential ingredient of human culture. Therefore, it has always been a major challenge to exploit recent technical achievements to assist, or replace, human skills in the field of music. For example, in the past, mechanical devices such as musical clocks were developed. The interesting question nowadays is to exploit electronic devices, especially the enormous potential of computers, for such purposes. The use of electronic devices for storing played music and for rendering it again with adjustable loudness, amplification of the different frequencies, tempo and sounds, has long been state of the art. Moreover, the MIDI-technique allows storage of a piece of music in the form of a MIDI-file. This involves a list of commands that may be generated live while a musician plays a piece of music on a MIDI-instrument or may be derived directly from the score. To play the music, these commands are fed into a MIDI-player (consisting of a software MIDI-Player and a MIDI-instrument). The conventional MIDI-technique allows manipulation of these commands so that the general loudness, tempo and sound of the played notes can be adjusted. Moreover, it is possible to adjust these parameters individually note by note. Consequently it is, in principle, possible to create any desired interpretation of a piece of music that is given in the form of its score. However, this requires considerable experience and patience.

An interesting question is to what degree it is possible to automate the interpretation of music. Some approaches to this have already been published [1-8, 12-16]. Our approach differs with respect to the goals and the applied methods. Our starting point is that many fuzzy methods have been developed which allow modeling of human operator behavior in the control of a technical process [17-20]. Compared with other approaches, such as that based on neural networks or on case-based reasoning, the fuzzy approach has some advantages. It allows consideration of the two types of knowledge that determine the actions of a process operator: conscious knowledge that a process operator can express in words, and knowledge about which they are not conscious. These considerations suggest that there are many similarities between the actions of a process operator and a musician who interprets music. Therefore, considering the fuzzy modeling of the behavior of a process operator and of a musician, who interprets music, as essentially the same problem, we strive for progress in two directions (Fig. 1). On the one hand we want to exploit fuzzy methods that have proved to be successful in technical applications for music interpretation. We are interested in the question of how far-reaching the fuzzy approach is for these purposes.

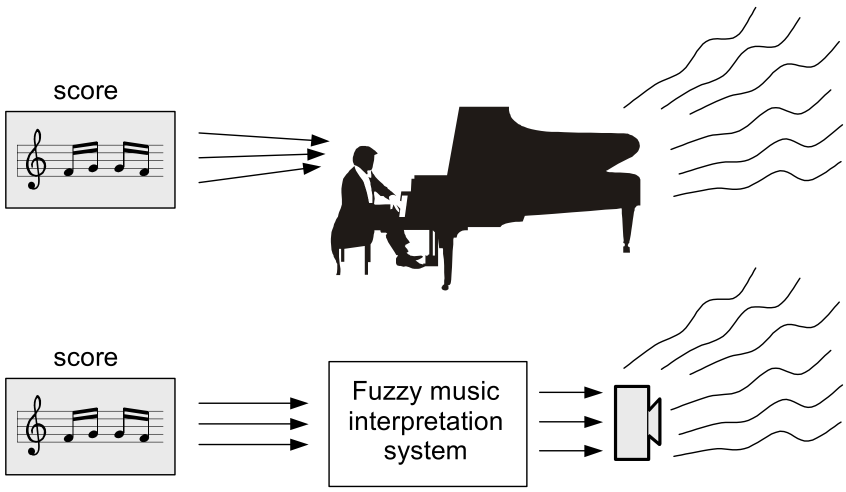


Figure 1: Pianist and fuzzy system interpreting a piece of music (top and bottom, respectively). Challenging questions: To what extent can the fuzzy system assist/replace the pianist? What further developments of fuzzy methods are meaningful for adequate interpretation of music?

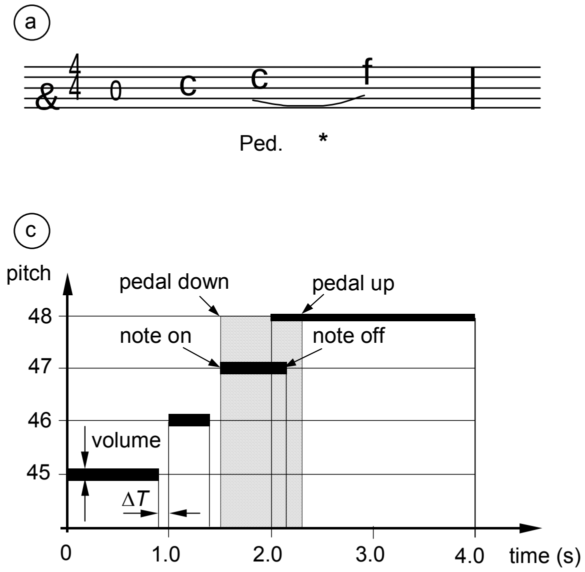
On the other hand we realize that the art of interpreting music has been developed and refined for centuries, much longer than the art of manually controlling an industrial process. Consequently, it has now reached a very sophisticated state. This makes the problem of generating a fuzzy music interpretation system an ideal test bed for demonstrating both the potential and the drawbacks of existing fuzzy methods.

Our paper is organized as follows. In section 2 we summarize the facts required for the MIDI-technique. In section 3 we present our fuzzy system, which allows interpretation of piano music that is given in the form of a score. We have focused on piano music to limit the complexity of the problem. We only have to manipulate the volume and the timing of the beginning and ending of each tone, not its sound. We show that in this application, conventional defuzzification strategies have an inherent deficiency and show how this is overcome using an advanced defuzzification strategy. In section 4 we outline possible refinements and further developments of our system. Finally, our conclusions are presented in section 5.

2 Use of MIDI-data

First, when speaking about music we mean sounds. There are at least two formats we use to work with sounds: audio and MIDI-data. Audio data, such as is stored on a CD, contains all the physical information (all the amplitudes of all the frequencies) associated with a sound. In contrast to this, MIDI-data can be considered as commands that are used for activating sounds on electronic music instruments. We consider only MIDI-data in this study.

Structure of MIDI-data



b

global commands
tempo, bar and key signature,...

local commands

number	ΔT(i) ms	action: note	action: pedal	attribute 1: pitch	attribute 2: volume	attribute 2: pedal intensity
1	0	on		45	6F	
2	900	off		45	0	
3	100	on		46	6F	
4	400	off		46	0	
5	100	on		47	6F	
6	0		down	–		7F
7	500	on		48	5F	
8	150	off		47	0	
9	150		up	–		0
10	1700	off		48	0	

Figure 2: Score (a), corresponding MIDI-data (b), and visualization of the MIDI-data (c).

Figure 2 (a) shows a musical score. The corresponding MIDI-data (Fig. 2 (b)) are composed of global commands (which are valid for the whole piece) mainly concerning the tempo and the key signature, and an ordered list of locally effective commands, each of which designates an action (note on, note off, pedal on and pedal off) together with the specification of certain attributes. The local commands define the channel for which the command is valid (not depicted in Fig. 2 (b)). Furthermore, they define, in the case of a note command, the pitch and the volume of the note, and in the case of the pedal command, the intensity of the pedal. The value of ΔT defines the time interval that has to elapse after the execution of the preceding local command until the current local command is executed. Figure 2 (c) is a visualization of the above local MIDI-commands.

Generation of MIDI-data and MIDI-playing

We can distinguish between the following methods of generating a MIDI-file, which contains the MIDI-data.

- The notes can be entered via a graphical user interface onto the note-line (*editing the score*). It is a similar procedure to a music composer writing notes on a sheet of paper. Once the writing is finished, the composer can save the work as a MIDI-file. Many extra features can be entered by this method of editing (e.g., crescendo, accelerando, pedal, ppp).
- Another method is to play the notes directly on a MIDI-instrument so that a musician can hear what he or she is recording (*recording the notes*). Two modes are possible: step-by-step recording, where the notes are entered one by one, and “live-recording”, where the notes are played in real time.

MIDI-playing means rendering a MIDI-file on a MIDI-instrument. Once a MIDI-file is created, one can only change the main tempo, the main volume for each MIDI-channel, and the MIDI-effects (e.g., glissando for a violin) for each channel while playing.

MIDI-elaboration of an interpretation

Often a musician receives a MIDI-file that is inadequate for his or her taste: either the presented interpretation is unsatisfying or it is a “mechanically played” piece of music (such as those we find in waiting-loops or in games). In this case, the musician can modify the MIDI-file manually. He or she can input, for example, global commands that apply global fixed or random variations of volume and tempo, or such variations individually, note by note.

Obviously the elaboration of an interpretation using existing MIDI-facilities requires considerable expenditure. We have to modify the volume, the time of the start and end of each note individually, either manually note by note, or by playing the piece repeatedly until the interpretation is satisfactory.

Our goal is to exploit fuzzy methods to provide much more comfortable and transparent means to generate a satisfactory interpretation automatically, or to elaborate it interactively.

3 The General Scheme

Figure 3 shows the general scheme of our system. The input is the information contained in a score (block 1). The outputs are MIDI-files that can be fed into a MIDI-player (not depicted in Fig. 3). Details are explained below.

3.1 Hardware and Software Components

For the system depicted in Fig. 3, we use a PC and the software tool DORA. This Windows-oriented software covers a broad spectrum of features that serve for the design and the online-use of controllers [21]. Compared with other existing tools, such as MATLAB/Simulink, DORA allows the design of considerably more flexible and more advanced fuzzy systems, such as fuzzy systems with TOR defuzzification, defuzzification by inference filter, and two-way fuzzy systems, and allows processing not only of conventional recommending (positive) rules but also of negative rules that express warnings and/or prohibitions [17, 18]. For rendering (MIDI-playing) of MIDI-files we use software MIDI-player (we realized it in PASCAL), a MIDI-interface on the PC (soundcard) and a MIDI-electronic piano.

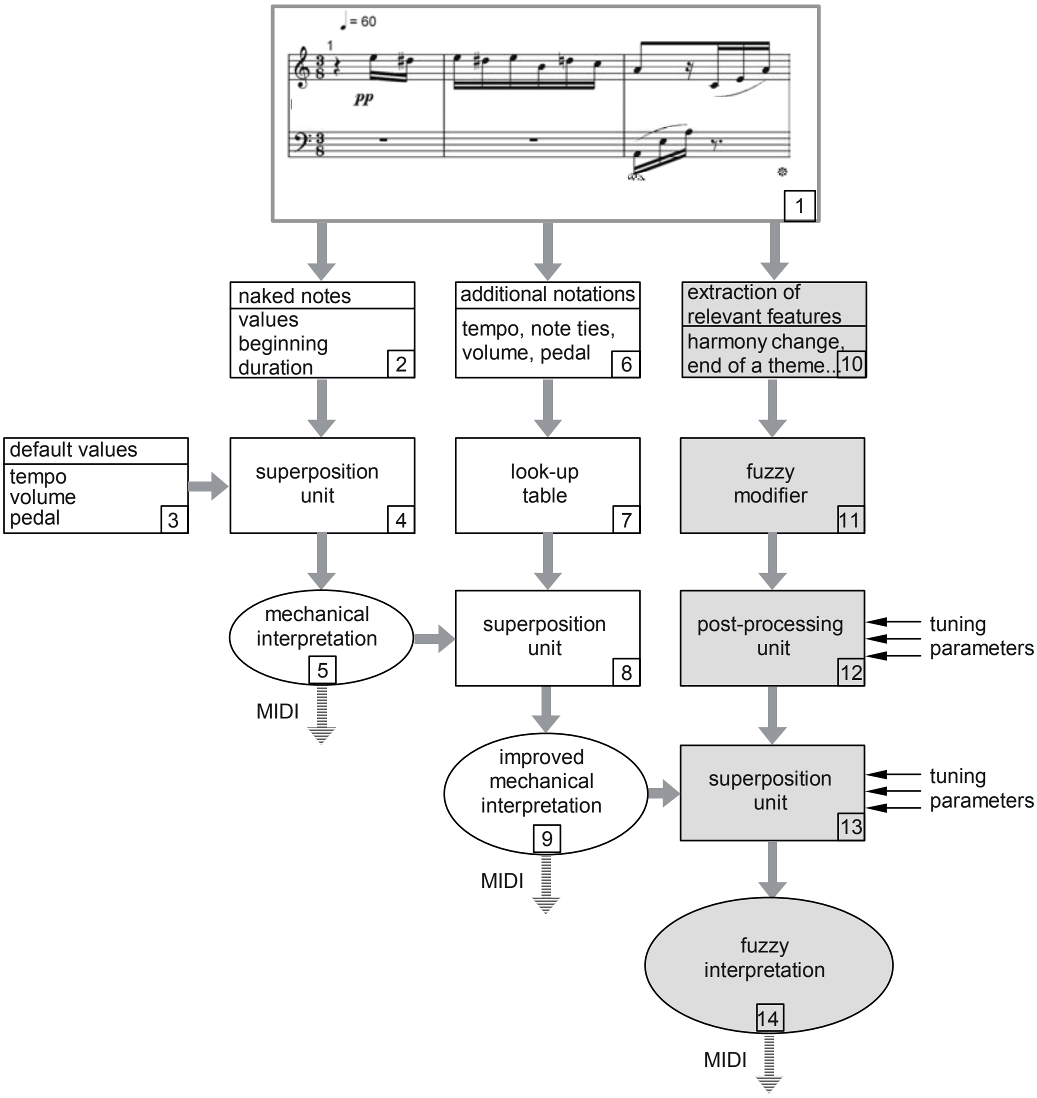


Figure 3: System for generating MIDI-files that represent a mechanical, an improved mechanical, and a fuzzy interpretation of a piece of music. The input of the system is the score of the piece of music.

3.2 The Temporal Variables of Notes

A piece of music can be considered as an ordered list of notes that is characterized by certain attributes such as the note *pitch*, *volume,* and the values of two time values. The two time values define the time of the start and the end of the note. We modify these time values by the manipulation of two other time values. The value *tempo* defines the number of milliseconds per standard note (e.g., a quarter of a note or a whole note). The value *duration* defines the duration of the note, expressed as a fraction of the value tempo. If we set, for instance, for all notes the same value *tempo*, each note-beginning is at a precise moment in the time base. In this case the note-endings can still be varied individually by manipulating the value *duration*.

If the duration of an individual note is reduced – say from 150 ms to 100 ms – the beginning of all subsequent notes will be 50 ms earlier, but their tempos and durations will not be affected.

By manipulating the duration value, legato can be realized: for this to occur, a note must end exactly where the next note begins. We also provide the option *superlegato*, which means that a note ends a little after the next note begins. To realize staccato, a note ends considerably before the next note begins.

3.3 Mechanical Interpretation

Our starting point is a piece of music that is laid down in the score. We encode the score and feed it into the PC. Here, we consider the composition "Für Elise" by L. v. Beethoven. The first bars are shown in Fig. 3, block 1. The encoded information for the piece of music is divided into two sections. The first refers to the "naked" notes and specifies the pitches (e.g., *e*, *d*, *a*, ...), and the time of the start and the duration of each note (compare with Fig. 2). The second section, derived from additional notations in the score, supplies additional information concerning tempo, note ties, legato, non-legato, staccato, volume and pedal. In the first step of information processing we consider only the first section of the information (Fig. 3, block 2) and superimpose it with default values for tempo, volume and pedal (Fig. 3, blocks 3 and 4). As a result, a MIDI-file is generated (Fig. 3, block 5) that represents a mechanical interpretation: It is played by the MIDI-player in a monotonous manner, without variations of tempo and volume (Fig. 4, top).

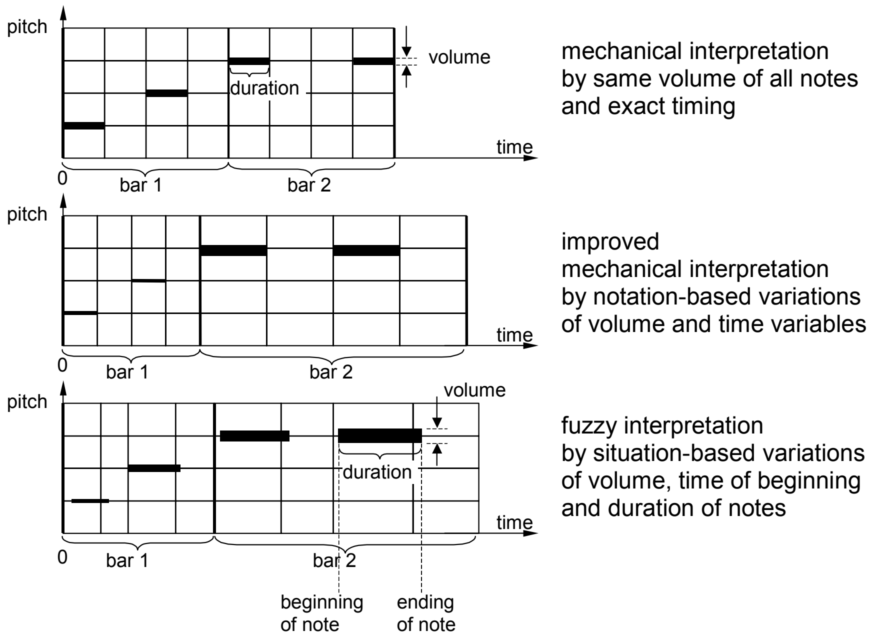


Figure 4: Mechanical, improved mechanical, and fuzzy interpretation

3.4 Improved Mechanical Interpretation

In the second step of information processing, an additional information block (Fig. 3, block 6) is used to modify the mechanical interpretation. We use a selectable look-up table (Fig. 3, block 7), which translates each symbol type of the additional notation for the whole piece into a fixed modification of the mechanical interpretation. Superimposing these modifications (Fig. 3, block 8) to the mechanical interpretation, we obtain a MIDI-file (Fig. 3, block 9), which represents an improved mechanical interpretation (Fig. 4, middle). This means we can now hear all music symbols that are explicitly contained in the score. However, it is still not music that would be played by a human pianist.

3.5 Fuzzy Interpretation

To improve the interpretation, we introduce situation-dependent modifications. For this we have consulted music experts and the music literature and established a set of recommendations concerning the interpretation of classical piano music in the form of 150 qualitative rules, such as:

IF < the position of a short right-hand note is close before a change from "major" to "minor" >
THEN < reduce tempo considerably > (1)

IF < right-hand note is part of a long decreasing sequence of notes AND the position of this note is in the middle of the theme >
THEN < reduce volume a little > (2)

IF < there is no pedal symbol AND there is a pause in left hand >
THEN < release pedal > (3)

These rules have the general form

IF < condition > THEN < recommendation > (4)

The conditions, which specify the rule premises, refer to certain context-dependent features or values of variables that are not necessarily explicitly visible in the score. The conclusions of the rules recommend modifications of the volume, of the tempo, and of the duration of the individual notes, and of the activation, the intensity and the releasing of the pedal (compare with Fig. 4).

To use such rules, we have to specify what is meant quantitatively by the qualitative linguistic values, such as *short*, *long*, *middle* or *a little*. For this, we use the concept of fuzzy membership functions. It allows us to express to what degree μ_k, with $0 \le \mu_k \le 1$, the premise of a rule R_k is met. The value μ_k (called degree of activation of rule *k*) determines to what degree the recommendation of the rule *k* is taken into account in the superposition of the recommendations of all activated rules. The following points are essential for our approach:

- We do not provide rules such as "bar 3 has to be played *piano*", but rather we use general rules. Each rule influences the interpretation globally (with respect to many bars) and also makes sense for other compositions of related style.

- The use of context-based conditions in the rule premises, in combination with the fuzziness (softness) of the membership functions, allows the production of sensitive modifications of interpretation.

The desired fuzzy interpretation is generated by the following information processing steps. First, the encoded information of the score is pre-processed in order to evaluate the values of predefined variables/features that are considered in the rule premises (Fig. 3, block 10, and Fig. 5). Second, these values are fed into the fuzzy modifier (Fig. 3, block 11). This is a fuzzy system (Fig. 1, left) consisting of a fuzzification unit, the rule base, an inference machine, and a defuzzification unit. Third, the output values of the fuzzy modifier are refined by a post-processing unit (Fig. 3, block 12): it exhibits transparent tuning parameters that allow interactive adjustment of the output values of the fuzzy modifier, live by ear, to satisfy individual taste. Finally, the output values of the post-processing unit are superimposed (Fig. 3, block 13) on the improved mechanical interpretation. In addition, the superposition unit exhibits transparent parameters for live tuning. As a result, we obtain the fuzzy interpretation (Fig. 3, block 14).

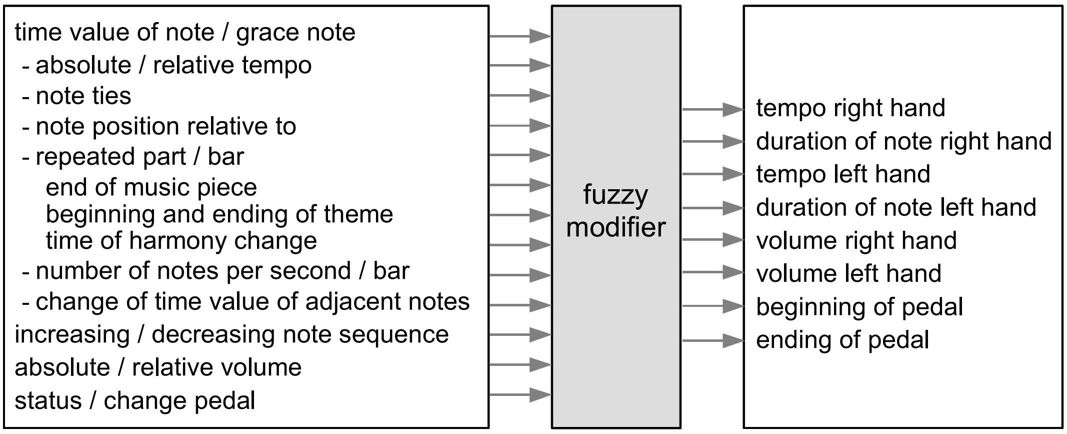


Figure 5: Input and output variables of the fuzzy modifier.

4 Details and Refinements

4.1 Membership Functions

For the modeling of each linguistic input (output) value we use up to four (10) membership functions in the form of overlapping triangles or trapezoids (singletons). We can modify the distribution of the input and output membership functions interactively for tuning purposes, beginning with equally distributed membership functions. However, as this tuning option requires adjustment of a large number of parameters and does not work online, we prefer to use the tuning parameters of the post-processing and superposition units (sections 4.3 and 4.4).

4.2 Defuzzification

Let the output membership function referring to an output value u, produced by all activated rules, be given by r singletons, $\{\mu_i, u_i\}$, $i = 1, 2, ..., r$, where u_i is the position and μ_i the activation of the i-th singleton:

$$\mu(u) = \begin{cases} \mu_i, & \text{if } u = u_i \\ 0, & \text{otherwise} \end{cases} . \qquad (5)$$

The prescriptions

$$u_{MAX} = u_j \text{ if } \mu_j > \mu_i \text{ for all } i \neq j \qquad (6)$$

and

$$u_{COG} = \frac{\sum_{i=1}^{r} \mu_i u_i}{\sum_{i=1}^{r} \mu_i} \qquad (7)$$

are the most common strategies for deriving an unequivocal real output value from $\mu(u)$. The first, the MAX (maximum) defuzzification, selects the best supported (most recommended) value u. The second, the COG (center of gravity) defuzzification gives a compromise.

This music interpretation problem shows that both the MAX and COG defuzzifications have an essential limitation. Let the fully activated rule R_1 produce the output singleton $\{\mu_1 = 1, u_1 = 2\}$ that recommends to the degree $\mu_2 = 1$, a *medium* $(u_1 = 2)$ increase of the volume for some bars. Let the partly activated rule R_2 produce the output membership function $\{\mu_2 = 0.8, u_2 = 1\}$, which recommends, to the degree 0.8, a *small* $(u_2 = 1)$ increase of the volume at the beginning of a bar (Fig. 6). Then it is obvious that if both rules are activated simultaneously, the resulting increase of volume should be *greater than medium*. More generally, here we need a defuzzification strategy that superimposes equi-directional recommendations of the individual rules so as to amplify each other. Neither the MAX nor the COG defuzzifications have this property. These considerations suggested the introduction of the TOR (torque) defuzzification

$$u_{TOR} = p \sum_{i-1}^{r} \mu_i u_i \qquad (8)$$

where p is a scaling factor [17, 18]. Figure 6 illustrates that this defuzzification has this desired property. (For p = 1, the resulting output value u_{TOR} corresponds to the torque induced by the masses μ_i considering their positive or negative distances from the neutral point u = 0.) This is the reason we use the TOR defuzzification method predominantly.

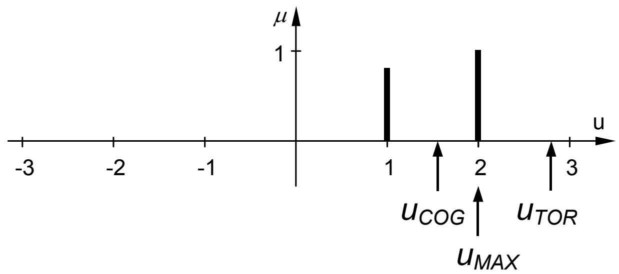


Figure 6: Defuzzification of an output membership function representing two equi-directional recommendations of two rules. The values $u_{COG} = 1.55$, $u_{\max} = 2$ and $u_{TOR} = 2.8$ are obtained by the COG, MAX, and TOR defuzzifications, respectively.

4.3 Post-processing Unit

The fuzzy modifier is designed so that the value range for each output variable is normalized within the range $-1 \le u \le +1$. Here, $u > 0$ means that the volume (or the tempo, or the duration) of a right-hand (or left-hand) note should be increased ($u < 0$ means that it should be decreased).

The post-processing unit transforms the output values u of the fuzzy modifier into the value

$$\hat{u} = \begin{cases} \varepsilon(a) + b \quad \dfrac{1-\exp(-c\,u)}{1-\exp(-c)} & \text{if } u \ge 0 \\ \varepsilon(a) + -d \;\; \dfrac{1-\exp(c\,u)}{1-\exp(-c)} & \text{if } u \le 0 \end{cases} \tag{9}$$

where a, b, c, d are tuning parameters with nonnegative values that are specified individually for each output variable of the fuzzy modifier. Here, $\varepsilon(a)$ is a random variable that allows the production of random variations of volume, tempo, and duration of the notes. The amount of these variations may be adjusted via the parameter a. We provide this random option because a pianist will never be able to control his or her playing perfectly to accord with his or her desired interpretation. Marked random variations of volume and time values will always be present to a greater or lesser extent.

The values b and d define the value range for $\hat{u}$. In the case of $\varepsilon(a) \equiv 0$, this is given by $-d \le \hat{u} \le b$. Consequently, we can adjust via b and d the general intensity of the influence of the fuzzy modifier. The value d determines the slope of the function $\hat{u}(u)$, especially for small values of u. Thus, we can adjust to what extent small variations of u produce noticeable (hearable) effects.

4.4 Superposition Unit

In the MIDI-data, the volume of a tone is encoded by an integer variable k with $0 \le k \le 127$. We superimpose these values k of the improved mechanical interpretation and the output $\hat{u}$ of the post-processing unit using

$$\hat{k} = \begin{cases} Int[k + \dfrac{\hat{u}}{b} q\,(127-k)] & \text{if } \hat{u} > 0 \\ 0 & \text{if } \hat{u} = 0 \\ Int[k + \dfrac{\hat{u}}{d} q\,k] & \text{if } \hat{u} < 0 \end{cases} \tag{10}$$

where $Int[z]$ rounds the element z to the nearest integer less than or equal to z. Here, q is a tuning parameter with $0 \le q \le 1$, which allows adjustment of the intensity of the influence of the fuzzy modifier.

We encode the tempo of a tone by positive values x (milliseconds). These values x of the improved mechanical interpretation are superimposed with the corresponding outputs $\hat{u}$ of the post-processing unit according to

$$\hat{x} = 10^{\frac{\hat{u}}{p}} x \qquad (11)$$

where p is a positive tuning parameter. This approach provides that the increase and decrease of x, which are produced by the values $\pm u$, are symmetrical on a logarithmic scale of x.

Finally, we encode the duration of a tone by positive values y with $0 \le y \le 1$ (ratio of the nominal duration of the note). These values y of the improved mechanical interpretation are superimposed with the corresponding outputs $\hat{u}$ of the post-processing unit according to

$$\hat{y} = Min\left(1.1,\, 10^{\frac{\hat{u}}{p}} y \right) \qquad (12)$$

where p is a positive tuning parameter. (Here we allow $\hat{y} > 1$ to realize *superlegato,* which means that a preceding note ends a little after the next note begins.)

5 Ongoing and Possible Refinements

5.1 Improved Tuning Unit

The approach (9), (10) works satisfactorily, however we consider it a first estimate. It should be refined by suitable experiments. Furthermore, the value range of the variable $\hat{u}$ may possibly be restricted in a more natural manner by using the Einstein sum

$$a \oplus b = \frac{a+b}{1+\frac{ab}{c^2}} \,. \qquad (13)$$

to determine the total activation of each singleton at position u_i (by summing the individual contributions produced by all activated rules that have the same conclusion) and to evaluate the sum (8).

5.2 Use of Negative Rules

The type (4) rules discussed so far are positive rules that produce recommendations. We process these rules as usual by a conventional Mamdani fuzzy system. More transparency for the processing of qualitative knowledge is obtained if negative rules in the form of

$$\text{IF} < condition > \quad \text{THEN} \quad < warning / veto > \qquad (14)$$

are also provided. These can be processed together with the positive rules by a two-way fuzzy system with hyperinference [17, 18]. It is well known that a conventional Mamdani fuzzy system is a universal approximator, and consequently allows production of any desired input/output characteristic. Therefore, the use of negative rules, in principle, will not lead to new characteristics. Essentially the use of negative rules provides more transparency in the processing of qualitative knowledge.

For instance, suppose that in the interactive process of refining the interpretation we have designed a set of 20 positive rules that recommend certain situation-dependent modifications of the volume. Let us further assume that we want the volume variation not to be too large for notes that belong to the theme. If we provide positive rules only, we can realize this in principle by adding the condition "AND note does not belong to the theme" to the premises of all 20 positive rules, and set up additional positive rules for notes belonging to the theme. However, instead of this costly and non-transparent procedure, we can leave the original 20 positive rules untouched and add only one single negative rule:

IF < note belongs to the theme >
THEN < u = *large* is FORBIDDEN >. (15)

If this rule is fully activated, its output membership function is given by $\mu^-(u) = \mu_{large}(u)$, where $\mu_{large}(u)$ is the membership function that models the linguistic value *large* (Fig. 7, left). Let $\mu^+(u)$ be the output membership function (in the form of singletons) produced by all activated positive rules. Then, the hyperinference produces from $\mu^+(u)$ and $\mu^-(u)$ a membership function $\mu(u)$ given by $\mu(u) = \mu^+(u) \wedge \neg\mu^-(u)$, or resulting from the strong or the weak veto strategy [17, 18]. Choosing the weak veto hyperinference, we obtain a resulting output $\mu(u)$ where singletons $\{\mu_i, u_i\}$ of $\mu^+(u)$ with large values of $|\mu_i|$ are suppressed as shown in Fig. 7, right.

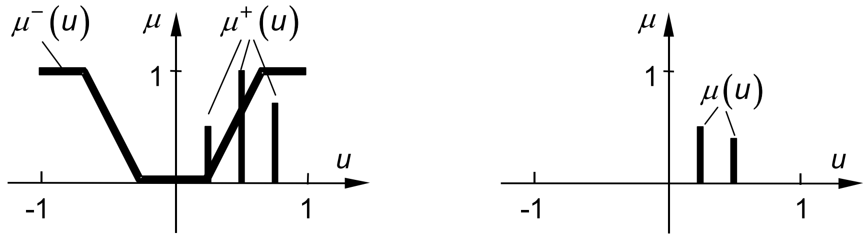


Figure 7: Output membership functions $\mu^+(u)$ and $\mu^-(u)$ produced by positive rules and one negative rule, respectively (left). Membership function $\mu(u)$ resulting from processing $\mu^+(u)$ and $\mu^-(u)$ by the weak-veto hyperinference (right). The application of the TOR defuzzification to $\mu(u)$ produces a smaller output value, compared with applying it to $\mu^+(u)$.

Applying the TOR defuzzification, the volume variations for notes in the theme are reduced compared with a situation where we have the same output $\mu^+(u)$ of the positive rules and no activation of the negative rule. What we wish to stress here, is that the use of negative rules supplies a much more transparent means of modifying the interpretation than if we are restricted to the use of positive rules only.

5.3 Data-based Approach

So far, we have set up the rules by use of expert knowledge. However, this knowledge-based approach only allows exploitation of that knowledge of which a musician is conscious. However, the interpretation of a good pianist is not only created by conscious knowledge, but also by unconscious emotions. In this respect there are similarities between a pianist and a human process operator who manually controls a technical process, partly by applying conscious rules, partly by following his or her unconscious experience. To access this unconscious knowledge, data-based methods for rule generation have been developed [19, 20]. These methods have proved successful in many technical applications.

Given a piece of music in the form of a MIDI-file that results from the live performance of a musician, these methods should allow extraction of the applied interpretation rules. Given the human interpretation in the form of a CD, such data-based rule extraction is far more difficult, but not *a priori* utopian. In both cases, an essential problem is to find relevant input variables for the rules, input variables that allow the specification of situations that are relevant for interpretation purposes. So far, we have chosen these variables considering the knowledge of experts (compare with Fig. 5). In the problem of modeling the musical interpretation of a musician, it may be that one has no idea on which features the interpretation is based. To overcome this problem, data-based methods may be applied that allow specification of relevant sets of variables if a large number of potentially relevant variables is given [22, 23].

6 Conclusions and Related Works

We presented "Für Elise" performed by our system to a class of music students. The music was rated as "a good performance of a good pupil who has had piano lessons for about four years". We consider this to be encouraging. The state now reached is satisfactory for trivial music purposes such as telephone waiting-loops or computer games. For such applications it is of interest that our fuzzy interpretation does not require charges for the authorship of the interpretation, in contrast to a human's interpretation. Naturally, for more demanding purposes, there is still considerable work to be done. However, we wish to stress the following points. We designed the rules considering mainly "Für Elise". However, the application of these rules to "Alla Turka" by W. A. Mozart also produces an acceptable interpretation. This means, it is possible to set up general rules, which can be applied to many pieces of music of a similar style. With our system, people who have musical inspiration but are not able to play an instrument, especially inexperienced and handicapped persons, can work out an interpretation and so express their emotions. Moreover, the system can be used for lessons. It allows demonstration of how a piece of music sounds if a specific rule of interpretation is applied.

Comparing our approach for automating the interpretation of music with already existing methods [1-7, 13-16], we conclude that the main goal of all the works mentioned is to create an interpreted classical music that could be played by humans. Due to the authors' biases, music for saxophone [3, 4], piano [1, 14-16], and violin [12] have been investigated. As far as we know, the most successfully generated music systems [1-4, 14-16] are based on collected musical examples, performed by a musician. The intention of others has been to construct rules that convert the written score, complemented with special symbols and marks, to a musically acceptable performance via a musical synthesizer [6]. The musical "facts" are also available via spectral analysis [3, 4], statistical methods (autocorrelation) [5, 7], and MIDI-format [16]. This information, together with musical rules chosen from music theory [8-11] and the experience of the authors, allows similar cases to be found in the collected examples, and based on these cases, a new piece of music to be modified [4, 16].

Our approach is different. We start from the score, and set up general rules that create situation-dependent variations of the volume, the tempo and the duration of the notes. It is essential that we use so-called incomplete rules, which means that each rule considers specific aspects such as “note is part of the theme”. Consequently, usually several rules, which consider quite different aspects, are activated simultaneously and contribute to the interpretation of the note considered. The advantage of this approach is that with few rules we can describe a huge number of different situations. For example 20 rules, which may be either fully activated or not activated, allow us to differentiate between 2^{20} situations. Consequently, we believe that our approach, based on incomplete rules, supplies more insight and allows a more transparent interaction with a human than a case-based approach as in [1-4, 14-16], which consider each case (situation) as a whole. The price we have to pay for our approach is that the superposition of the contributions of the different rules has to be more sophisticated. A mere averaging, as in the case-based approaches, is inadequate.

It is interesting to realize that our system may serve for data compression. Note that the total opus of Bach, together with many interpretation rules, fits on one single CD. Instead of the current situation of only being able to vary the volume and the relation of high and low frequencies of the tones, people may select the desired mode of interpretation. For example, the style of Horowitz or of Volodos, or a selectable mixture of both, may be chosen and adapted to the listener’s taste by activating predesigned rules.

We consider it an open question as to what extent and how quickly these goals will be reached. In any case, the complexity of these problems will induce further developments of fuzzy modeling strategies such as in the creation of TOR defuzzification.

References

[1] P. Blashill: The Creative Professor, Inc. The Magazine for Growing Companies, September 2001

[2] C. Cope: Computer Modeling of Musical Intelligence in AMI. Computer Music Journal, 16 (2), pp. 69 – 83, 1992

[3] R. L. de Màntaras and J. L. Arcos: The Synthesis of Expressive Music: Challenging CBR Application. In: ICCBR 2001, LNAI 2080, pp. 16 – 26, Springer Verlag, Berlin, Heidelberg, 2001

[4] R. L. de Màntaras and J. L. Arcos: An Interactive Case-Based Reasoning Approach for Generating Expressive Music. Applied Intelligence, 14(1), Southwest Texas State University, USA, Kluwer Academic Publishers, pp. 115 – 129, January-February, 2001

[5] P. Desani and S. de Vos: Autocorrelation and Study of Musical Expression. In: Proceedings of the 1990 International Computer Musical Conference, San Francisco, 1990

[6] A. Friberg: Generative Rules for Music Performance: A Formal Description of a Rule System. Computer Music Journal, 15 (2), pp. 56 – 71, 1991

[7] H. Katayose: Expression Extraction in Virtuoso Music Performances. In: Proceedings of the Tenth International Conference on Pattern Recognition, Washington DC, IEEE Computer Society Press, 1990

[8] F. Lerdahl and R. Jackendoff: A Generative Theory of Tonal Music. The MIT Press Series on Cognitive Theory and Mental Representation, Cambridge, MA, 1983

[9] F. Lerdahl and R. Jackendoff: An Overview of Hierarchical Structure in Music. In: Stephan M. Schwanaver and David A. Levitt, editors, Machine Models of Music, MIT Press, Cambridge, MA, pp. 289 – 312, 1993

[10] E. Narmour: Beyond Schenkerism. University of Chicago Press, 1997

[11] E. Narmour: The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model. University of Chicago Press, 1990

[12] G. de Poli, A. Rodá and A. Vidolin: Note-by-Note Analysis of the Influence of Epressive Intentions and Musical Structure in Violin Performance. New Music Research, 27 (3), pp. 293 – 321, 1998

[13] J. Sunberg, A. Askenfelt and L. Fryde: Musical Performance: A Synthesis-by-Rule Approach. Computer Music Journal, 7 (1), pp. 37 – 43, 1983

[14] G. Widmer: Modelling the Rational Basis of Musical Expression. Computer Music Journal, 19 (2), pp. 76 – 96, 1995

[15] G. Widmer: Applications of Machine Learning to Musical Research: Empirical Investigations into the Phenomenon of Musical Expression. In: R. S. Michalski, I. Bratko and M. Kubat editors, Machine Learning and Data Mining: Methods and Applications, Wiley, Chichester, UK, 1998

[16] G. Widmer: Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report. AI Communication, 14 (3), pp. 149 – 162, 2001

[17] H. Kiendl: Fuzzy Control methodenorientiert. Oldenbourg-Verlag, München, 1997

[18] H. Kiendl: Decision Analysis by Advanced Fuzzy Systems. In: J. Kacprzyk, L. Zadeh, editors, Computing with Words in Information/Intelligent Systems, pp. 223 – 242, Physica-Verlag, Heidelberg, 1999

[19] T. Slawinski, A. Krone, P. Krause and H. Kiendl: The Fuzzy-ROSA Method: A Statistically Motivated Fuzzy Approach for Data-Based Generation of Small Interpretable Rule Bases in High-Dimensional Search Spaces. In: M. Last, A. Kandel, H. Bunke, editors, Data Mining and Computational Intelligence, pp. 141 – 166, Physica-Verlag, Heidelberg, 2001

[20] T. Slawinski, J. Praczyk, U. Schwane, A. Krone and H. Kiendl. Data-based Generation of Fuzzy Rules for Classification, Prediction and Control with the Fuzzy-ROSA Method. In: European Control Congress (ECC'99), Karlsruhe, volume CD-ROM, 1999

[21] A. Krone, P. Krause, T. Slawinski and R. Knicker: WINROSA 2.0 and DORA for Windows 6.2. In: Symposium Identification (SYSID) 2000, Santa Barbara, USA, pp. 521 – 526, Elsevier Science, Amsterdam, The Netherlands, 2000

[22] D. Schauten, B. Nicolaus and H. Kiendl: Evolutionäres Verfahren zur Selektion relevanter Merkmalssätze für die datenbasierte Fuzzy-Modellierung. In: Proceedings 11. Workshop Fuzzy-Control des GMA-FA. 5.22, Universität Dortmund, 2001. VDI/VDE GMA-FA. 5.22, Wissenschaftliche Berichte Forschungszentrum Karlsruhe, pp. 133 – 147, 2001

[23] D. Schauten, B. Nicolaus and H. Kiendl: An Evolutionary Concept for Selecting Relevant Sets of Input Variables for Data-Based Fuzzy Modeling. In: Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems (EUNITE '01), Puetro de la Cruz, Tenerife, Spain, December 13-14, 2001