

Chapter 8

Human–Machine Musical Composition in Real-Time Based on Emotions Through a Fuzzy Logic Approach

Pedro Lucas, Efraín Astudillo and Enrique Peláez

In this chapter, a method for representing human emotions is proposed in the context of musical composition, which is used to artificially generate musical melodies through fuzzy logic. A real-time prototype system, for human–machine musical compositions, was also implemented to test this approach, using the emotional intentions captured from a human musician and later used to artificially compose and perform melodies accompanying a human artist while playing the chords. The proposed method was tested with listeners in an experiment with the purpose of verifying if the musical pieces, artificially created, produced emotions in them and if those emotions matched with the emotional intentions captured from the human composer.

8.1 Introduction

Emotions are particularly intrinsic to music composition and performance despite composers might have or not considered them while composing. Vickhoff [13] argued that we do not have control over our emotions, because they are triggered involuntarily and non-consciously by nature. This raised the scientists' interest about finding ways of modeling emotions computationally to drive them in musical composition.

P. Lucas (✉) · E. Astudillo · E. Peláez
Facultad de Ingeniería en Electricidad y Computación, Centro de Tecnologías
de Información, ESPOL, Guayaquil, Ecuador
e-mail: pepaluca@espol.edu.ec

E. Astudillo
e-mail: ejastudi@espol.edu.ec

E. Peláez
e-mail: epelaez@espol.edu.ec

Bezirganyan [2] showed that a particular melody could provoke a variety of emotions to different listeners at the same time. Vickhoff [13] also showed that an emotion could be perceived distinctly by different listeners depending on who they are and the situation involved; findings which are relevant for a system when composing melodies and pretends to provoke similar effects on listeners according to the level of emotions that a human composer would.

This work is intended to help musicians in their creative process of composing musical pieces, but with a better understanding about the role emotions play in music, through a set of synthetic intelligent partners. In contrast with previous work presented in this introduction, our approach relates the corpus of melodies with the emotions that could produce on people, considering a real-time environment.

8.2 Background

This section discusses previous work related to musical composition based on emotions, and fuzzy logic applied to music.

8.2.1 Approaches for Music Composition Based on Emotions

There have been different approaches for developing systems that analyze the emotions' content on music composition. Xiao Hu et al. [8] developed *Moody*, a system that classifies and recommends songs to users based on the mood they want to express or have in that particular moment, a solution that considers the use of support vector machines and a Naive Bayes classifier.

Strapparava et al. [11] showed that music and lyrics are able to embody deep emotions. They proposed syntactic trees for relating music and lyrics, annotated with emotions on each lyric line; in this case, support vector machines were used to classify and demonstrate that musical features and lyrics can be related emotionally.

Suiter [12] proposes a novel method using concepts of fuzzy logic to represent a set of elements and rules, considering expressiveness to trace a trajectory of musical details related to composition and establishing important points for the application of fuzzy logic over musical parameters; also, Palaniappan et al. [4] used fuzzy logic to represent musical knowledge, in which a fuzzy classifier was a component of a system for knowledge acquisition intended to Carnatic musical melodies.

Xiao Hu [7] and Wiczorkowska et al. [14] used emotions as labels for organizing, searching, and accessing musical information, whereas Misztal et al. [9] exposed a different approach by extracting emotions content from text, which then are used as inspiration for generating poems. The system proposed expresses its feelings in the form of a poem according to the affective content of the text.

8.2.2 *Emotions*

Finding a proper definition of emotion has been controversial and a notorious problem [3, 10]. Biologists and neurologists differ in their definition and both refer to it as a subjective quality of our present state. Emotions, according to biologists, are an important steering mechanism for animals and humans. Neurologists believe that conscious observation of emotion is specific to humans [13].

Scherer [10] claims that emotions are a reproduction of various events produced by an external or internal stimuli. Those events could be measured by taking in consideration the main following aspects.

1. Continuous changes in appraisal processes at all levels of the central nervous system,
2. Motivational changes produced by the appraisal results,
3. Patterns of facial and vocal expression as well as body movements,
4. Nature of the subjectively experienced feeling state that reflects all of these component changes.

X. Hu. [6] in his work mentions 5 fundamental generalizations of mood and their relation with music, which tell us that:

1. Mood effect in music does exist.
2. Not all moods are equally likely to be aroused by listening to music.
3. There do exist uniform mood effects among different people.
4. Not all types of moods have the same level of agreement among listeners.
5. There is some relation between listeners' judgments on mood and musical parameters such as tempo, dynamics, rhythm, timbre, articulation, pitch, mode, tone attacks, and harmony.

8.2.3 *Fuzzy Logic in Music*

Palaniappan et al. [4] proposed a knowledge acquisition method using a *fuzzy classifier* with the goal of representing patterns, which then could be used to generate style-based music. In this case, the notes from melodic samples are analyzed and their membership degrees are obtained by the occurrences of *established patterns* on each sample.

Suiter [12] also proposed to relate fuzzy logic principles with musical elements. In this work, the elements to represent knowledge are non-linear parameters like timbre, rhythm, frequency, and amplitude more than linear elements like notes. The focus is on expressiveness, where fuzzy sets are managed through a fuzzy controller.

In this work, fuzzy logic is used to represent *melody patterns* where emotions are denoted as fuzzy sets with membership degrees in the interval [0, 100], where 0 means absence of feelings associated to an emotion, and 100 a complete match of a feeling regarding an emotion, these values are subjective and assigned based on

a human perception. Therefore, each melody pattern is labeled with emotions and their corresponding membership degree, will represent an *emotional intention* that we will use in the fuzzification–defuzzification process.

8.3 Compositional Model Based on Emotions

The proposed model for musical composition will be used for producing melodies artificially and accompanying a human artist during a musical performance.

8.3.1 Architecture for Musical Knowledge Elicitation and Representation

The architecture for musical knowledge elicitation and representation proposed in [1], was designed based on the criteria of two experimental musicians and algorithmic compositional methods, as illustrated in Fig. 8.1, which emphasizes musical composition through a fuzzy logic approach. This architecture considers a *knowledge base* that is composed by transition matrices, obtained through a *Markov chain process* over melodies provided by human musicians and labeled with emotions by them.

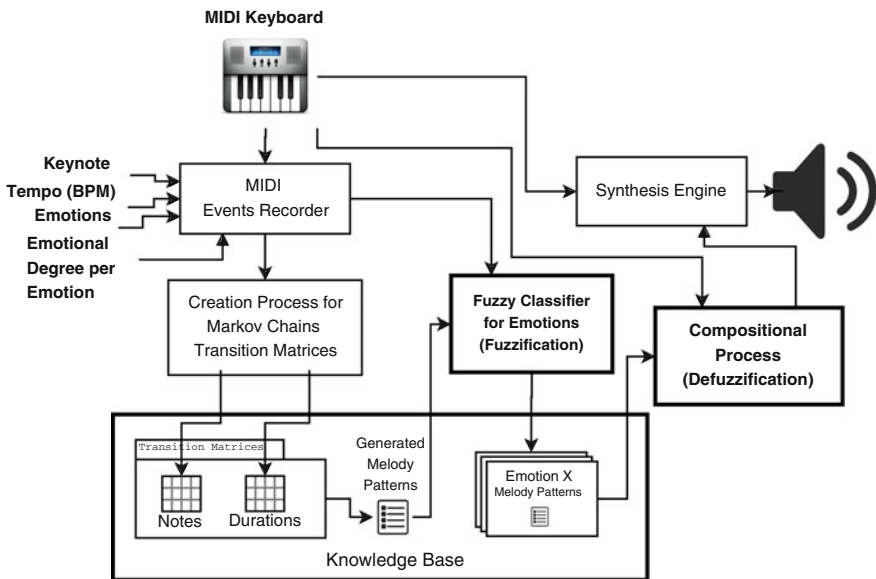


Fig. 8.1 Knowledge elicitation and representation, with compositional process

From these matrices, new melody patterns are generated and emotions are assigned through a *fuzzy classifier* implemented with a *fuzzification process* as described in Sect. 8.3.2. These patterns are then stored in the *knowledge base* for later use in the *compositional process* that produces musical pieces which are played back by a *synthesis engine* through the speakers.

Based on this architecture, a software prototype was implemented. A human musician trained the system for nurturing the *knowledge base* by performing melodies and providing the corresponding entries as depicted in Fig. 8.1. The input parameters, *emotions* and *emotional degree per emotion*, represent the *emotional intention* the musician wants to provoke to the audience; for example, happiness(80), sadness(10), and serenity(75) are three emotions weighted by the musician to express, between 0 and 100, different intentions to produce a particular emotion (0 represents no intention and 100 represents an absolute intention to produce the emotion).

The *knowledge base* is nurtured with the melody patterns generated by the composition algorithm described in [1], which is based on Markov chains. As shown in Fig. 8.1, the melody patterns generated are then labeled with the emotions and their intentions, which are defined by the musician, through a *fuzzification process* as described in Sect. 8.3.2.

For the compositional process, a real-time system was developed to support a *musical human–machine improvisation*, in which the musician (human) plays the chords for a piece that is *composed while is playing*, during this process the system (machine) produces melodies that accompany those chords by “remembering” the *musical data from chords* in the past (*notes and durations*) and stored in the *knowledge base*; also, the musician provides the *emotional intentions* previous to the composition; that is, the *emotions* that wants to produce in the audience with the *emotional degrees*, as in the training process. All this data is used to select the right melody patterns saved in the *knowledge base*, through a *defuzzification process*, as described in Sect. 8.3.2, and produce music.

8.3.2 Fuzzy Logic Approach

Considering the emotional influence by music over humans, a linguistic variable called *emotions* will be used to capture the emotions, which are fuzzy sets that the musician provides to the model, as well as the corresponding weights to represent the *emotional intentions*; therefore, the melodies provided by the musician to train the model, will create a set of solutions (melody patterns) that can be recalled later for music composition in real-time using the provided emotions.

The approach for musical composition entails two main processes. First, a fuzzification process which allows the classification of melody patterns; second a defuzzification process, which selects the piece of melody that is played back during real-time composition.

Fuzzification Process: Classification Method

The melody patterns are generated applying Markov chains as described in the architecture shown in Fig. 8.1 [1], and represented using Eq. (8.1), where *note* is an integer between 0 and 127, representing a MIDI number for the musical note, and *duration* is a relative time that is based on the *tempo* (beats per minute, BPM) which marks the rhythm for the melody pattern.

$$\begin{aligned} \text{Melody Pattern} : & (note_0, duration_0), \\ & (note_1, duration_1), \\ & \dots, (note_n, duration_n) \end{aligned} \quad (8.1)$$

The relative time durations ($duration_i$) that were described above, are taken from this fixed array of float numbers [0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4, 4.25, 4.5, 4.75], which are representations for musical durations [5], where 1 is a *quarter note* ♩, as a reference for obtaining other durations.

The subscript n from (8.1) represents the size and it is an input parameter that controls the number of generated pairs (*note*, *duration*) that will compose the melody pattern. In this approach, *musical rests* [5] (intervals of silence) are not considered in the melody pattern because they are merged with their immediate previous note in order to reduce the complexity of the representation for the pattern.

The transition matrices are built from the set of melodies played by a human composer who provides the emotions and their corresponding emotional intentions, these matrices do not use these intentions in the process of generating new melody patterns, but they are used during the classification process, as a source for labeling those new patterns of size n , with the emotions given by the musician, as described below:

1. The process is applied to notes and their durations in an independent way, so a melody pattern is split in two arrays; one for *notes* and the other for *durations*. An x melody pattern generated from Markov chains will be named as MM_x (melody machine), and the corresponding arrays are $MMnotes_x$ and $MMdurations_x$
2. The melodies recorded by humans, have a size m , and will be named as MH_y (melody human) then, the two arrays will be split, following the previous procedure; hence, we have $MHnotes_y$ and $MHdurations_y$. The next steps will get the distance between MM_x and MH_y , which will be used in the emotion's labeling for MM_x . An example for these first steps is showed in Fig. 8.2.
3. To calculate the difference between each element of the $MMnotes_x$ and $MHnotes_y$ matrices, which contain MIDI notes numbers between 0 and 127, we use the following equation:

$$\Delta note_{ij} = |MHnotes_y[i] - MMnotes_x[j]| \text{ mod } 12, \quad (8.2)$$

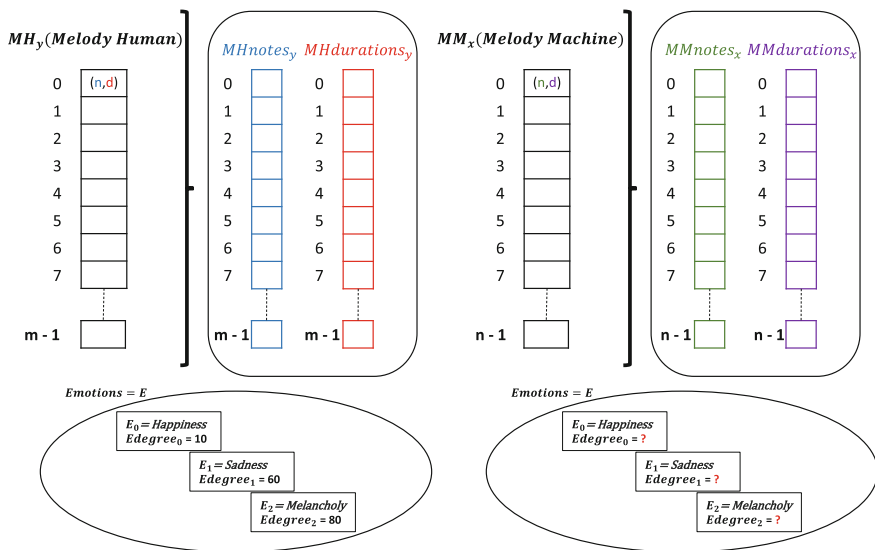


Fig. 8.2 Example corresponding to the structure used in melody patterns for human and machine

These are musical notes, which are linear musical representations composed essentially by 12 elements distributed in several octaves [5], so in this equation the octaves are not relevant because of the mod operation.

- Equation (8.2) will be applied for each element of $MMnotes_x$ and $MHnotes_y$ to calculate the distance between $MMnotes_x$ and a segment $S_y[k]$. The segment $S_y[k]$ is a subset of size n , where k is positive integer between 0 and $m - n$, and is included in $MHnotes_y$. If $m \geq n$, there will be $m - n + 1$ segments contained in a melody created by a human, but if $m < n$, then there will be just one segment and the operations will not consider the entire $MMnotes_x$ array. Equations (8.3) and (8.4) are used to calculate the distance between $MMnotes_x$ and a segment $S_y[k]$. Figure 8.3 shows the representation of segments for $m \geq n$ and Fig. 8.4 illustrates how the distance between $MMnotes_x$ and a segment $S_y[0]$ is obtained; in this case, each $\Delta note_{ik}$ is calculated as an average, to get the required distance $ds[0]$, which in this example is 7, 00.

$$\Delta note_{ik} = |MHnotes_y[i + k] - MMnotes_x[i]| \text{ mod } 12, \quad (8.3)$$

$$d(MMnotes_x, S_y[k]) = ds[k] = \frac{\sum_{i=0}^{\min(m,n)-1} \Delta note_{ik}}{\min(m, n)}, \quad (8.4)$$

This model considers an average among the $\Delta note_{ik}$ values in a segment $S_y[k]$, whose results will always be a number between 0 and 11, due to the mod operation.

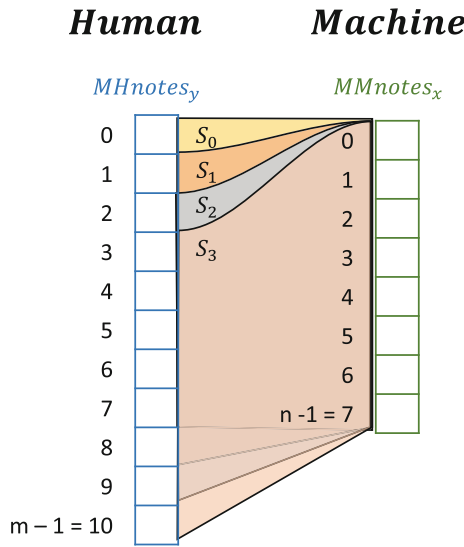


Fig. 8.3 Representation for segments in melody patterns

<i>Human</i>		<i>Machine</i>		
<i>MHnotes_y</i>		<i>MMnotes_x</i>		For segment <i>S₀</i> :
0	76	0	72	$\Delta note_{00} = 76 - 72 \bmod 12 = 4$
1	81	1	71	$\Delta note_{10} = 81 - 71 \bmod 12 = 10$
2	76	2	60	$\Delta note_{20} = 76 - 60 \bmod 12 = 4$
3	81	3	62	$\Delta note_{30} = 81 - 62 \bmod 12 = 7$
4	83	4	60	$\Delta note_{40} = 83 - 60 \bmod 12 = 11$
5	81	5	62	$\Delta note_{50} = 81 - 62 \bmod 12 = 7$
6	80	6	60	$\Delta note_{60} = 80 - 60 \bmod 12 = 8$
7	81	7	64	$\Delta note_{70} = 81 - 64 \bmod 12 = 5$
8	83			
9	84			
$m - 1 = 10$				

$$ds[0] = \frac{\sum_{i=0}^{\min(11,8)-1} \Delta note_{i0}}{\min(11,8)} = 7,00$$

Fig. 8.4 Example for calculating for distance between segment 0 and a machine melody pattern

- The number of distances obtained for each segment, between *MMnotes_x* and *MHnotes_y* is $m - n + 1$; hence, to determine the minimum distance from all segments in *MHnotes_y* and *MMnotes_x*, we use the following equation:
For $m > n$,

$$d(MMnotes_x, MHnotes_y) = dnotes_{xy} = \min\{ds[k] : k \in [0, m - n + 1]\}, \quad (8.5)$$

For $m < n$ there will be just one distance, that is, $d(MMnotes_x, MHnotes_y) = ds[0]$.

6. This calculation of the distance between a generated melody $MMnotes_x$ and a human melody $MHnotes_y$, has to be applied to all melodies in the *knowledge base*; therefore, if p is the number of human melodies in the *knowledge base*, then the *closest human melody*, in terms of distance and the related pattern, to $MMnotes_x$ is given by Eqs. (8.6) and (8.7).

$$DNotesMin_x = \min\{dnotes_{xy} : y \in [0, p - 1]\}, \quad (8.6)$$

$$MHnotes_{min} = MHnotes_y, \text{ such that: } \min\{dnotes_{xy} : y \in [0, p - 1]\}, \quad (8.7)$$

7. The human melody for notes $MHnotes_{min}$ is related with a complete melody (notes and durations) that we called $NotesMH_{min}$, which was one of the previous melodies that were labeled by a musician who established emotions and its corresponding weights along the training. This set of emotions will be denoted as E and have a size that we will call ne with a specific emotion E_r , such that r is an integer in the interval $[0, ne - 1]$. For each human melody MH , there is a set of emotions E with their corresponding weights w_r . Thus, the emotions and its weights for $NotesMH_{min}$ are used for labeling the new generated pattern $MMnotes_x$ as described in Eq. (8.8).

$$EMMnotesX_r = EMHnotesMin_r \left(1 - \frac{DNotesMin_x}{11}\right), \quad (8.8)$$

We use 11 to normalize the minimum distance $DNotesMin_x$ for each pattern, because this value is in a range between 0 and 11, as described in step 4) of this process.

To assign weights to each emotion $EMMnotesX_r$ in the melody machine patterns $MMnotes_x$ we use Eq. (8.8) where the emotions weights, that label the *closest human melody to $MMnotes_x$* , come from $NotesMH_{min}$. These emotions weights are denoted as $EMHnotesMin_r$ and are weighted by $DNotesMin_x$ as the equation describes. For example, if a generated pattern ($MMnotes_x$) has a distance of 3.5 ($DNotesMin_x$) regarding its nearest human melody ($NotesMH_{min}$), and the emotions given by the musician to that melody are *happiness*(10), *sadness*(90), and *melancholy*(75) (given that $ne = 3$ and r is in $[0, 2]$), then the generated pattern will be weighted using $EMHnotesMin_r(1 - \frac{3.5}{11})$; therefore, the results are *happiness*(6.82), *sadness*(61.36), and *melancholy*(51.14), for that generated pattern ($MMnotes_x$).

8. This process is extrapolated to *durations*; therefore, the same equations are applied to the array $MMdurs_x$, but considering: First, the difference between elements of $MMdurs_x$ and $MHdurs_y$, which will be given by Eq. (8.9).

$$\Delta dur_{ij} = \min(|MHdurs_y[i] - MMdurs_x[j]|, 4), \quad (8.9)$$

Because durations are relative to the *tempo* (BPM), the value is fixed to 4 beats, which allows us to have a numeric reference for the normalization factor when the emotion weights are calculated. It is 4 because a *complete rhythm measure* (bar) can be basically marked as 4 beats like a metronome does [5]; and, second, the normalization factor of 4, as described before.

9. Finally, the process has to be applied to all *MM* patterns; that is, to all the generated patterns produced by the Markov chains component. Therefore, the *knowledge base* will have two kinds of weighted sets, one for notes *MMnotes* and another for the durations *MMdurs*. Since these sets are not merged in a weighted *MM*, a melody can be built with notes and durations that come from different generated patterns when the *defuzzification process* acts, providing more flexibility in the compositional process.

Defuzzification Process: Compositional Method

As in real-time musical composition (*improvisation*) [5], a *human-machine musical composition* takes place when a human musician plays the chords for a musical piece and it is accompanied by the machine or viceversa. The inputs involved are described below:

Start Input Before initializing the system, the human musician must provide the weights for the intended emotions. These values are related to the emotional intention that the human composer wants to transmit to the audience. Also, the *tempo* (BPM) and the *keynote* have to be given.

Real-Time Input While the musician is playing, the artificial agent gets the musical notes generated by the artist (*chords*), and produces new melodies in real time, using the data stored at the *knowledge base*.

For the compositional process, we use a metronome to guide the human composer. On every beat marked by the metronome, the system produces a melody pattern that is the result from the compositional process. This process uses the acquired notes from the previous period of time between the current beat and its predecessor, as shown in Fig. 8.5.

Not all notes of a generated melody pattern of size n are played, because of the overlapping notes in every beat. This overlapping will produce dissonance if the playing notes are still being performed in the next chord, which means that, the musician execution might not be congruent with the last set of notes [5]. To solve this problem, the melody pattern only is executed in a random number of notes between 1 and the *inputs size* (notes played by human composer). For example, if we have a melody pattern with $n = 20$ and it is received in a specific time, an input like this (48, 50, 52, 55), in MIDI notation, represents the notes (C3, D3, E3, G3), then the system will split the melody pattern in 1, 2, 3 or 4 notes. However, if the human composer plays a lot of notes, all notes will be reproduced. This behavior produces an interesting effect that makes the system generate harmonies more than melodies along the composition; an effect that is not dissonant.

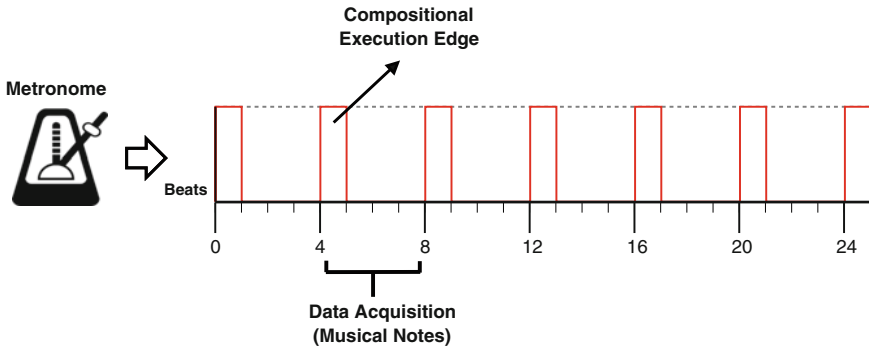


Fig. 8.5 Metronome model for data acquisition and compositional execution

To choose the melody pattern that best fits on every beat, we follow the next defuzzification procedure:

1. Since the *knowledge base* can be very wide, we need a strategy to search the best solution according to the input. Hence, the generated melody patterns are organized in *balanced binary trees*.
2. In this approach, the *knowledge base* is structured in *ne* balanced binary trees, such that *ne* is the number of emotions that are involved for all the generated patterns, where all these patterns have a certain membership degree per emotion. Thus, each tree will represent an emotion where the **keys** are the *emotional degrees* (weights or membership degrees) and the **values** are the *melody patterns* associated with that emotion. Although this representation requires some extra memory, because of the keys, it is a worthy trade-off because it helps to reduce the search space to find the right patterns and reproduce them, as we will see later.

For example, if the *knowledge base* is trained with three emotions *happiness*, *sadness*, and *melancholy*, then we have three balanced binary trees, as illustrated in Fig. 8.6. Then, each melody pattern is added to each of those trees, based on its emotion and emotional degree; that is, if the degree of happiness is 20.0, then a new node is created in the *happiness tree* with a *key* = 20.0, and, if there is already a node with that key, then the pattern *P* is associated to a set of patterns which belongs to that node, in order to share this same node as in Fig. 8.7. Since the *knowledge base* have two kinds of weighted melody patterns, one for *notes* and other for *durations*, we have to consider six binary trees in this example.

3. The emotions weights provided by the musician in the *start input* might not be registered as keys in the trees, because it could not have appeared previously in the training; for example if the musician enters *happiness(10)*, *sadness(90)*, and *melancholy(75)*, and there is not the key 10 in the tree for happiness, then we are going to consider the *nearer keys* as shown in Fig. 8.8.

The **nearer keys** represent the nodes that meet these requirements:

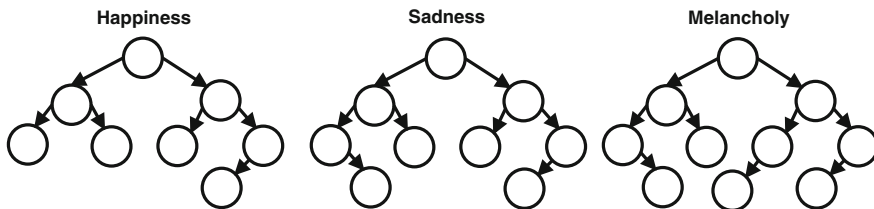


Fig. 8.6 Emotional trees examples

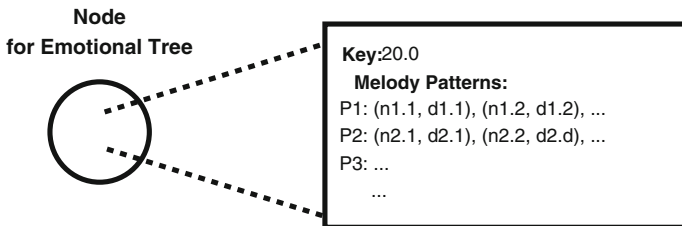
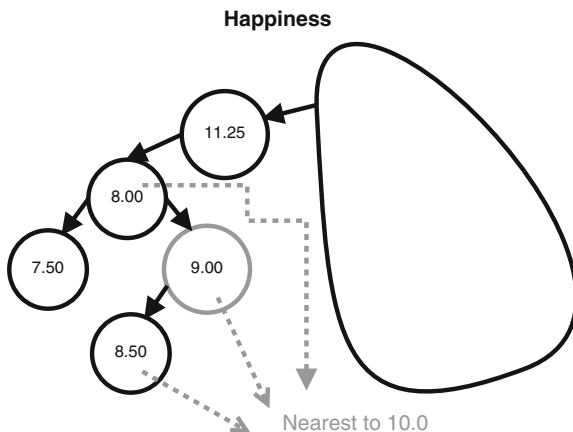


Fig. 8.7 Node structure example for an emotional tree

Fig. 8.8 Nearer keys to 10 from a happiness tree



- If the target emotion weight is found, then we will collect all the melody patterns associated with this target node and its adjacent nodes; that is, the parent and its children.
- If the target is not found, we will traverse the tree until a null leaf is found, we will go up to its parent and as before, we will collect all the melody patterns from this node and from its adjacent nodes.

The objective is to have a reduced solution space with the patterns that matters for each emotion independently, which are closer to the emotion’s weights given by

the musician in the *start input*. This procedure can take place during the system initialization.

- When the process is running and the human composer is playing, the system receives the notes from a MIDI keyboard (real-time input). This input and the reduced solution space trees are used for playing a melody through the speakers. To get a melody pattern, the system iterates over the reduced solution space looking for patterns whose *notes* meet the following two criteria:

- The set of weighted emotions Eh of size ne , which the human performer gave at the initialization time, is compared against all the weighted emotions Em for each pattern in the reduced space by using Eq. (8.10), which is a *Euclidean distance* to take into account all the emotions we use.

$$Dhm = \sqrt{\sum_{r=1}^{ne} (Em_r - Eh_r)^2}, \tag{8.10}$$

The goal is to find the melody pattern that has the **minimum emotional distance** and also is *musically consistent* with the input notes, as explained below:

- The melody pattern to be chosen must be *musically consistent* with the harmony (chords) that the human composer is playing. Hence, we use the following criteria: *If the first note in the candidate melody pattern is part of the input provided by the human composer, then that melody pattern is consistent with the received harmony.* There could be other heuristic criteria; however,

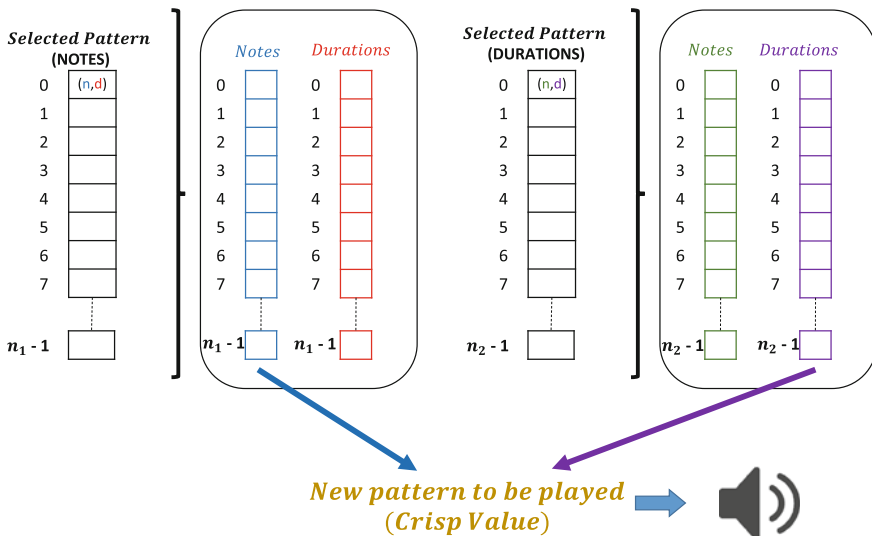


Fig. 8.9 Generation for the new melody pattern

we do not want to have a strong restriction that inhibits the artificial creativity of the system.

These two criteria are merged by an *and* (\wedge) operator to shape one expression and get the target melody pattern based only on *notes*. For *durations*, we just need the first criteria.

5. Finally, the two chosen arrays, *melody pattern for notes* and *melody pattern for durations* are putting together to generate a new melody. Therefore, from the fuzzy sets for *Emotions*, we get a crisp value (melody pattern) as in Fig. 8.9.

8.4 The Experiment for Musical Intention and Perceptions

8.4.1 Procedure

Musicians with academic background trained the system with 15 melodies with an average of 30.0 s per melody. The emotions selected by the artists were five: *happiness*, *serenity*, *sadness*, *nostalgia*, and *passionate*, which were weighted with emotional degrees between 0 and 100 for each melody, depending on the emotional intention, also the *keynote* and *tempo* were provided during the training. The system generated 30 melody patterns using Markov chains, which were weighted through the fuzzification process.

The human musician performed the harmonic base (chords) for 15 musical pieces approximately for 60 s, also provided the emotional intentions for each piece, such that the system played melodies that were consistent with the provided harmony and the emotional intentions using the defuzzification process.

These 15 musical pieces were then played to other people that listens western music, hence the pieces were weighted with the emotional perceptions as they were perceived, 30 people filled the assessment. A summary for this procedure is described in Fig. 8.10.

8.4.2 Results

The results presented in Fig. 8.11, as box plots, showed that for the musical piece 1, for example, the emotional intention from the musician and the perceived emotions by the listeners differed from each other; however, the perception tilts toward the emotions in a similar way as the intention. As seen in Fig. 8.11, there is more *serenity* and *nostalgia* than *happiness* and *passion* in the perceptions, as well as in the intentions, but *sadness* is not adjusted to this behavior. The plots from other musical pieces, for all emotions behaved similarly, or have one emotion that is not adjusted. Other emotions that were felt by the listeners, and not intended by the system where melancholy, reminiscence, calm, relaxation, depression, hope, and anxiety.

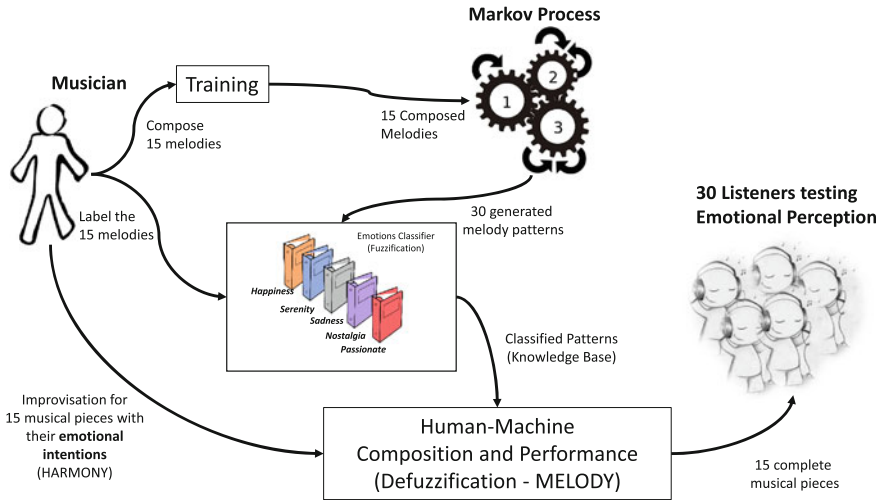


Fig. 8.10 Elements for the procedure and their interactions

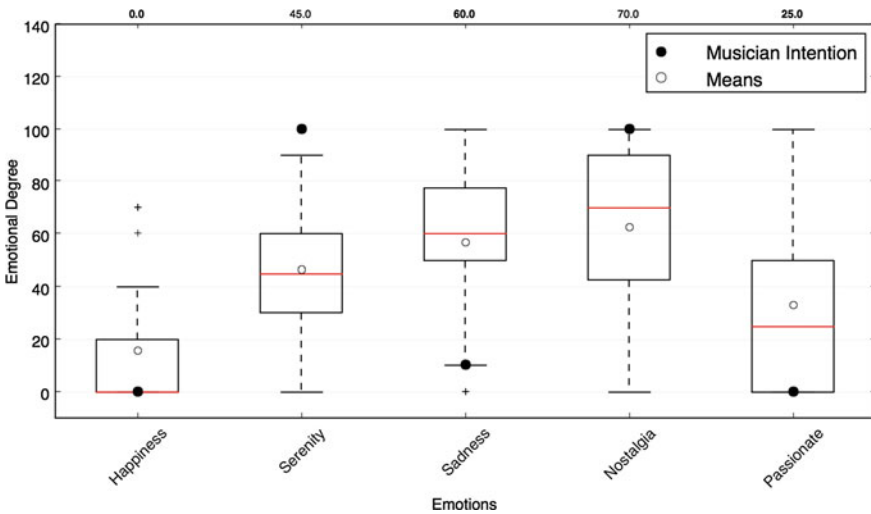


Fig. 8.11 Emotional intention and perceptions for Musical Piece 1

In Table 8.1, we present the results for a Levene’s test that is used to assess the equality of variances for a variable, calculated for two or more groups. In our case, the test is applied to every song, and the groups per song are the emotions (*happiness, serenity, sadness, nostalgia, and passionate*). This test was applied with 95% confidence and tell us that the variability for each song, regarding the emotions, does not differ significantly except for *Musical Piece 5* and *Musical Piece 6*, which means that listeners perceived each song with a similar degree of vagueness.

Table 8.1 Levene's test for the emotions' variances on each musical piece

Songs	W	P-value
Musical Piece 1	1.731	0.146
Musical Piece 2	0.377	0.824
Musical Piece 3	1.833	0.125
Musical Piece 4	1.353	0.253
Musical Piece 5	3.289	0.0129
Musical Piece 6	2.816	0.0274
Musical Piece 7	1.952	0.104
Musical Piece 8	0.753	0.557
Musical Piece 9	1.120	0.349
Musical Piece 10	0.124	0.973
Musical Piece 11	1.460	0.217
Musical Piece 12	0.968	0.427
Musical Piece 13	0.412	0.799
Musical Piece 14	0.978	0.421
Musical Piece 15	1.599	0.177

Finally, despite these results, listeners did not report any comment that suggested a random composition of melodies; though, they felt some pieces had similar melody patterns.

8.5 Conclusions

This paper presents a musical composition approach based on human emotions as fuzzy sets. The processes for fuzzification and defuzzification for these sets, were implemented in the context of a real-time system, that performed musical pieces along with human partners, who felt a well-timing execution from the artificial agent, which resulted in a proper synchronization between players, just like human musicians playing with an emotional connection. The human musicians reported that sometimes the emotional intention changed a little in order to perform consistently with the system; however, it did not affect the composition significantly, as stated by the musicians. Thus, the system is not restricted to what is required to produce, but it contributes with its own *style* to the compositional process.

This unexpected change of intentions could have caused that the emotional perception did not match significantly with the emotional intention as the results suggest. Also, this experiment did not control the emotional status for each listener, so it could have influenced in the answers; nevertheless, the variability regarding these answers is similar on each musical piece, which means that there is a subjectivity degree to be considered when people listen a song that avoids an expectation about the emotional

intention. However, all the listeners reported that they felt the emotions that were established for the testing, and even other distinct emotions. These results show that the proposed method does influence on people's feelings that listen Western music.

This research contributes to the creative compositional process, providing to musicians inspirational material that is generated from the same source from which the system is trained, a style that is indeed preferred by them, based on their knowledge during the process of composing music. This approach could also be applied to other areas where real-time multimedia applications are needed; such as, video games or interactive experiences that require dynamic sound design.

Acknowledgments We thank the musicians who contributed with its criteria and talent for the basis of this work. We also thank to all people who, with patience and enthusiasm, supported us in this research as listeners.

References

1. Astudillo E, Lucas P, Peláez E (2015) Adquisición del Conocimiento en el Proceso de Composición Musical en Base a Técnicas de Inteligencia Artificial. In: XI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento, pp 171–185. Riobamba
2. Bezirganyan AH (2004) Analysis and estimation of emotionally colouring of music performance. In: Proceedings of the 8th international conference on music perception and cognition, pp 710–712
3. Cowie R, Sussman N, Ben-Ze'ev A (2011) Emotion: concepts and definitions
4. Ganesh Ram S, Palaniappan CT, Ramakrishnan MS, Devanathan R (2004) Knowledge engineering of creative musical expressions using Carnatic music ideology. In: International florida artificial intelligence research society conference
5. Hewitt M (2008) Music theory for computer musicians. Course Technology PTR, Boston, USA
6. Hu X (2010) Improving music mood classification using lyrics, audio and social tags. ProQuest Dissertations and Theses, p 141
7. Hu X (2010) Music and mood: Where theory and reality meet. In: Proceedings of iConference, pp 1–8. <http://www.ideals.illinois.edu/handle/2142/14956>
8. Hu X, Sanghvi V, Vong B (2008) Moody: a web-based music mood classification and recommendation system. In: Proceedings of the music information retrieval, pp 3–3
9. Misztal J, Indurkha B, Poetry generation system with an emotional personality. Computationalcreativity. Net http://computationalcreativity.net/iccc2014/wp-content/uploads/2014/06/6.3_Misztal.pdf
10. Scherer KR (2005) What are emotions? And how can they be measured? doi:10.1177/0539018405058216
11. Strapparava C, Mihalcea R, Battocchi A (2007) A parallel corpus of music and lyrics annotated with emotions. In: LREC, pp 2343–2346
12. Suiter W (2010) Toward algorithmic composition of expression in music using fuzzy logic. In: Proceedings of the 2010 international conference on new interfaces for musical expression, pp 319–322
13. Vickhoff B (2008) A perspective theory of music perception and emotion, vol 90, 1st edn. Intellecta DocuSys. Västra Frölunda, Gothenburg
14. Wiczorkowska A, Synak P, Lewis R, Ras Z (2005) Extracting emotions from music data. Foundations of intelligent systems, pp 456–465. <http://www.springerlink.com/index/lrc0995xxl5m12x4.pdf>