

Real-Time Maqam Estimation Model in Max/MSP Configured for the Nāy

Fadi M. Al-Ghawanmeh¹, Mohammad T. Al-Ghawanmeh², Mohammad W. Abed¹

¹Music Department, University of Jordan, Amman, Jordan

²Music Department, Yarmouk University, Irbid, Jordan

Email: f_ghawanmeh@ju.edu.jo, ghawanmeh_mohd@yahoo.com, m.abed@ju.edu.jo

Received 25 December 2015; accepted 13 March 2016; published 16 March 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Automatic maqam estimation is considered significant toward improving multimedia live music performances and automatic accompaniment. This contribution proposed a real-time maqam estimation model developed in the visual programming language MAX/MSP and configured for the nāydukah. The model's design stood on basic formulas of Arab music maqamat as explained in theory and applied in practice. The model consisted of different layers of competition; the first was for the identification of the instant tonic of the melodic figure, and the second was for the recognition of its identifying E (E, E half-flat and E flat). Those two competitions were used to estimate the maqam in real-time. Then, accumulated estimation results were used to estimate the maqam in longer durations; five-second and full duration. The model was evaluated using professionally performed nāy improvisations. Results reflected a success in estimating all the studied maqamat when the full improvisation was considered. In addition, results were very good for real-time and five-second estimation where average estimation confidence was 75.98% and 80.04%, respectively.

Keywords

Real-Time Music Systems, Arab Music, Maqam Estimation, Nāy, Music Signal Processing

1. Introduction

This contribution proposed a real-time maqam estimation model configured for the nāy and based on basic formulas of the Arab maqamat (plural of maqam) as explained in theory and applied in practice. The article also presented an evaluation of this model when using nāy improvisations as input. To the best of our knowledge, this contribution is the first presenting a real-time maqam estimation model adapted and tested for an Arab instrument. It is worth pointing out that unlike occidental music, only narrow attention was paid—in literature as

well as in the industry—to computer-aided analysis of Arab music, whether the analysis of performances or the acoustics of instruments [1]. The mainstream practice of Arab music was influenced exhaustively; although 96% of traditional Jordanian songs were composed on maqamat having neutral intervals ($3/4$ pitch interval), only 13% of contemporary popular Arab songs are. In addition, over 99% of Arab popular songs broadcasted today on media are composed solely on five maqamat despite having a rich heritage of tonalities exceeding 100 maqamat [2].

Maqam estimation is considered significant because of several reasons. For example, it is an important and initial step toward providing an automatic accompaniment to performed music. Failing to find the right maqam may consequently lead to providing an automatic accompaniment composed in a wrong key. This will for sure decrease the accompaniment reliability and usefulness [3]. Real-time maqam finding can also be used in live performances to influence the visual components and effects in the performance venue. This is because each maqam has a particular general feeling or mood: happiness, sadness, spirituality, etc. So such feelings can be reflected on lights, colors, images, etc.

The nāy, sometimes written as Ney, is a handcrafted woodwind instrument of cane. It is used in the Arab world and in other regions such as Turkey and Iran. Nāyists usually have sets of seven different-length nāyāt (plural of nāy) to allow for several maqam transpositions. The most commonly used Nāy of the set is called the nāydokah. The nāy is shown in **Figure 1** and the range of the nāydokah is given in **Table 1**. The pitch range of this nāy covers from C4 to G6, skilled performers may also produce few higher notes. In each octave, it is possible to play two neutral intervals in addition to the twelve semitones. Some chromatic tones need the special performance skill of half-hole opening. Some performers prefer to replace this technique by adding two extra holes controlled by the pinky fingers. Some tones slip slightly over or below well-tempered tuning. The nāy cane is composed of nine fragments separated by eight nodes. The nāy's cavity has a tight waist solely at the very first node; the waist allows for performing the high tones when blowing into the embouchure hole is enforced. Blowing air into the instrument is made nearly vertically [4].

The nāy is an essential Arab instrument that may perform in solo, or as a basic instrument in the Arab takht ensemble that also includes qanoun, oud, violin and riq. The nāy is important in Arab orchestras as well [1]. In addition to the importance of this instrument in Arab music, our proposed maqam estimation model was configured for the nāyDokah for technical reasons as well. The nāy has a simple sinusoidal signal, and its tone-range is more discrete when compared to the oud or violin. Also, unlike other takht instruments, it is less common for the Nāydokah to play transpositions of a particular maqam. In this case, another nāy of different length is usually used [5]. All those reasons let pitch detection be less challenging, which allows for more confident explorations in maqam estimation.



Figure 1. The nāy [4]. (a) Front side; (b) Backside; (c) embouchure hole.

Table 1. Range of the nāy “dokah” [1].

First octave	Second octave	The rest of tones
C ₄	C ₅	C ₆
C [#] ₄ /D ^b ₄	C [#] ₅ /D ^b ₅	C [#] ₆ /D ^b ₆
D ₄	D ₅	D ₆
D [#] ₄ /E ^b ₄	D [#] ₅ /E ^b ₅	D [#] ₆ /E ^b ₆
E ₄	E ₅	E ₆
F ₄	F ₅	F ₆
F [#] ₄ /G ^b ₄	F [#] ₅ /G ^b ₅	F [#] ₆ /G ^b ₆
G ₄	G ₅	G ₆
G [#] ₄ /A ^b ₄	G [#] ₅ /A ^b ₅	
A ₄	A ₅	
A [#] ₄ /B ^b ₄	A [#] ₅ /B ^b ₅	
B ₄	B ₅	

The remaining of this article is organized as follows. Literature review is presented in Section 2. The theoretical background of the proposed model is overviewed in Section 3. The implementation of the model is discussed in Section 4. Evaluation and discussion are presented in section 5. Finally, in Section 6, we conclude this contribution and propose future work.

2. Literature Review

In [6], spectral analysis was applied to study the effect of the material of the Nāy instrument, reed or metal, on the acquired timbre. The analysis also tackled the number of segments in the instrument, and then the 9-segment choice of the Arab Nāy was spectrally justified. In [1] [7]-[9], time domain and spectral analysis were used to find and improve pitch detection and automatic music transcription of Nāy recordings. Such improvements were necessary to increase the efficiency of several educational and artistic applications such as melody analysis, automatic instrumental and vocal accompaniment and query-by-playing [10]. Improving the manufacturing and capabilities of the Nāy was discussed in [11].

Automatic instrumental accompaniment to Arab vocal improvisation was discussed in [3] and [12], and used for educational purposes in [13]. Furthermore, a web application delivering such service is available in [14]. Improving the technicality and the commercialization of such accompaniment models was tackled [15]. However, these contributions did not present any maqam finding model despite its importance to the success and usefulness of accompaniment applications.

A music scale or mode identifies a group of notes that are employed in composition and treated as a one set. Each scale is identified by the pitch intervals between the sequence of notes in one octave, and recurring in all upper or lower octaves [16] [17]. Arab maqamat are better to be studied and described in the context of the arab musical heritage, but the closest counterpart from the occidental music might be the mode [18], particularly, the Greek modes.

Several research contributions tackled the problem of finding the musical scale automatically from an audio signal. Some common names for this challenge are key finding [19], key detection [20] or key estimation [21]. Mainstream techniques applied in scale estimation are usually chroma-based and consist of two successive stages. In the first, the pitch profile of the audio signal is extracted, and then mapped in the second stage to a database of profiles, each represents a particular scale. The pitch profile, or chroma vector, consists of a weighed set of all available pitch classes (12 classes in classical western music). The weights of the classes are usually obtained as follows: First, Fast Fourier Transform, FFT, is found. Next, a constant-Q filter bank is applied to divide the spectrum into different zones, each belongs to a particular quantization step. Then, the energy of in each

bank is calculated. Finally, all banks returning to a particular pitch class are folded to produce a pitch weight [22].

When applying the aforementioned key estimation method, or slightly altered versions of it, scale estimation obtained some fair results when applied on instrumental occidental music, but results were never robust. Actually, scale estimation is yet a hot MIR subject [3]. Examples of related contributions include that of [23] who evaluated his off-line scale estimation system against a database from the western common practice repertoire. The reported accuracy ranged between 85.5% - 88.9%.

Applying scale estimation on Arab music can introduce more challenges because of several reasons such as having dozens of maqamat (plural of maqam): nine basic maqamat families with about 30 to 40 most used maqamat. Furthermore and unlike occidental music, neutral intervals (3/4) and microtonal subtleties are common in Arab music [18]. In [24], a chroma-based method was suggested to automatically classify traditional Turkish music. Similar to Arab music, the Turkish is essentially melodic and has lots of maqamat, and more available intervals as compared to classical occidental music. Their article concentrated on the classification of individual (solo) instrumental improvisation, taqasim. Performances were classified into nine basic maqamat. Their work achieved a partial success, and discussed three main sources of errors; the classifier, the used audio recordings and the gap, in some occasions, between theory on the one hand and practice on the other hand. In [25], there was an attempt to improve these results by considering the conventional melodic progressions, seyir or masâr, and by processing only the first quarter of the recording rather than the full duration. This achieved a sort of improvement to this off-line classification model.

In [26], the chroma-based technique was not used. The paper demonstrated an algorithm for scale estimation in real-time. The authors found the pitches with their strengths using the Fast Fourier Transform, and then they applied a particular algorithm for the generation of a center of effect. The approach was tested on a classical western polyphonic audio input, (produced from a MIDI source) and revealed promising results.

Max/MSP is a well-known visual programming language that can be used to create diverse performance, or educational real-time applications [27]. In [28], the Max/MSP environment was used to implement an experiment aimed at studying the correlation between the ethnicity and relative pitch identification. In [29], Java and Max/MSP was used to implement a real-time beat tracker that was aimed at keeping the synchronization between a drums performer and the electronic sequencer.

3. Theoretical Background of the Model

The first step toward the maqam estimation model was pitch detection. This task was fulfilled using the Max/MSP external object “fiddle~” [30] and [31]. This object used a frequency-domain approach to find the fundamental frequency. It received the wave signal, buffered a block of a certain size, and then outputted the midi number of the instant pitch along with other possible outputs in real-time. We tuned the block size experimentally and found that a 50 millisecond block gave good detection results, this finding went in correlation with the finding reported in reference [32].

The proposed maqam estimation model was based on four key ideas about Arab music. These are:

1. Arab maqamat (plural of maqam) are normally constructed of two successive tri-, tetra-, or penta-chords. The lower chord is more important and is called the trunk while the upper chord is called the branch [33].
2. There are nine essential maqamat [34] on which the overwhelming majority of today’s contemporary Arab songs are composed [2]. The lower chord of each essential maqam is unique and can be considered as an identifier to that maqam.
3. In Arab music, as a tonal music, the melodic sentence is more-likely to finish at the tonic of the tetra-chord.
4. Even though the smallest interval in Arab music is the quartertone, this interval cannot be a scale intervals. Rather, the 3/4 tone interval can. Ex., if the tone E half-flat (neutral E) is among the maqam’s tones, it is certain that neither E natural nor E flat is among the maqam notes.

Accordingly and for the most common maqamat, maqam estimation depends on finding its lower chord. This can be performed by firstly detecting the chord’s tonic, and then detecting the accidentals (flat, half flat, natural, etc.) of one or more identifying tones. e.g., the tonic and the tone E as an identifying tone are sufficient to classify the themaqam of a melody to one of the following basic maqamat: rast, nahawand, bayati, kurd, sikah and ajam (on C). **Figure 2** presents the lower tri-, tetra-, and penta-chord of these maqamat. Our model is configured to find the maqam of a melody performed on any of these maqamat.

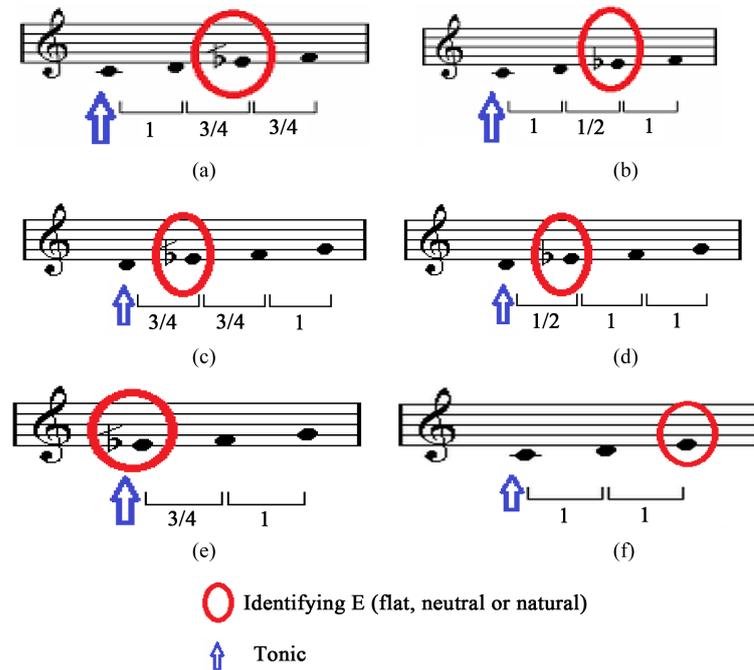


Figure 2. Six basic Arab tri-, tetra-, or penta-chords [18]. (a) rast; (b) nahawand; (c) bayati; (d) kurd; (e) sikah; (f) ajam (on C).

The lower step tone to E half flat (Ed) is D, and the upper step tone is F. This is the case for the maqamatrast, bayat, and hazam. However, pitch of Ed in bayati is slightly lower than in sikah or rast. This difference is equal to one Turkish kuma (or cumma). Each octave consists of 53 logarithmically equal kuma, and since the octave consists of 1200 logarithmically equal cents, the one kuma is about 22.6 cents [35]. So accordingly, the lower-step interval from Ed in rast and huzam is 7 kuma, and the upper step is 6, in bayat it is just the opposite. **Table 2** shows the Lower-step and upper-step intervals around the identifying E (E, Ed or Eb) in Turkish kuma in all the studied Arab maqamat. Based on these intervals, **Figure 3** presents a description of the range of each of the identifying notes (E, Ed or Eb). Those ranges are used in building the model as a Max/MSP patch.

4. Model Implementation

In this section we discuss the major parts of the maqam finding model as implemented in Max/MSP. In the following sub-sections, we present three layers of competitions; the first is for the identification of the instant tonic of the melodic figure, the second is for the recognition of its identifying E. Those two competitions are used to estimate the maqam in real-time. Afterwards, a third competition is considered to estimate the long-term maqam.

4.1. Competition for the Instant Tonic

Figure 4 depicts the patch fragment used to identify instant tonics of melodic figures. Calculations are performed continuously, but the final output is triggered by a “bang” that occurs only when a rest is detected for a minimum duration of 300 milliseconds, *i.e.*, when having at least six blocks of silence (or pitches out the nāy range) in a raw. This number, 300, was found experimentally while taking into consideration that the period separating two similar successive legato tones may reach the duration of 100 milliseconds. And, definitely, this period is not considered a silent note [9].

As shown in part (a) of **Figure 4**, pitch detection results (elements) of the last 11 blocks are buffered to the shift register “bucket”. Whenever a rest is detected, we know that the newer six elements (300 milliseconds) are silent elements. So the older 5 elements belong to the tonic, therefore. This is because the melodic sentence is most-likely to end at the tonic of the tetra-chord, as assumed earlier in this section. Those 5 elements are packed in one list, sorted and then the median (third element) is outputted. The sorting avoids outputting extreme values,

pitch name	D5		Eb5	Ed5 bayati rast		E5		F5
intervals (commas)		4		2	1	2		4
midi num.	74.00		74.91	75.36	75.58	76.04		76.94
Pitch range		74.46	Range of Eb	75.14	Range of Eb	75.81	Range of E	76.49

Figure 3. Description of the range of each of the identifying notes (E, Ed or Eb).

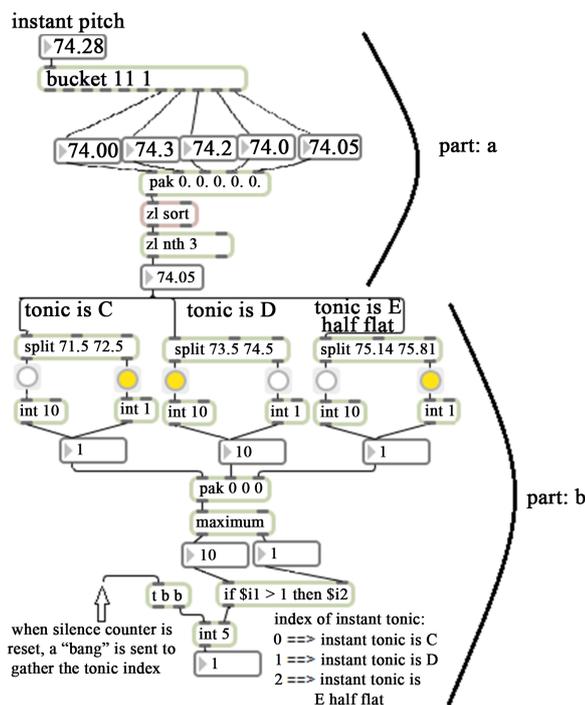


Figure 4. Max/MSP Patch fragment used to identify instant tonics of melodic figures.

Table 2. Lower-step and upper-step intervals around the identifying E (E, Ed or Eb) in Turkish kuma (cumma) in the studied Arab maqamat [36].

Rast		
lower step interval (D – Ed)		upper step interval (Ed – F)
7 kuma		6 kuma
bayati		
lower step interval (D – Ed)		upper step interval (Ed – F)
6 kuma		7 kuma
sukah		
lower step interval (D – Ed)		upper step interval (Ed – F)
7 kuma		6 kuma
nahawand		
lower step interval (D – Eb)		upper step interval (Eb – F)
4 kuma		9 kuma
kurd		
lower step interval (D – Eb)		upper step interval (Eb – F)
4 kuma		9 kuma
ajam		
		upper step interval (E – F)
9 kuma		4 kuma

and thus prevents considering a transient or noise element as a pitch. As appears in the figure, the midi number of the instant tonic at the snapshot moment was 74.05.

In the second part, see part (b) of **Figure 4**, we check if the midi number of the instant tonic falls within the range of any of those three choices: C, D or E half flat. Whenever this condition is met, the choice’s index number is stored in an “int” Max object, and is triggered whenever silence reaches 300 milliseconds. In the figure, the range of each choice is presented in a particular “split” object, and the index numbers of choices C, D or E half flat are 0, 1 and 2, respectively. As appears in the figure, the index of the instant tonic at the snapshot moment was 1.

4.2. Competition for the Identifying E

Figure 5 depicts the patch fragment used to recognize the identifying E. We make a competition to see how many instant pitches fall within the range of each of those three choices: E flat, E half flat and E natural. There is a counter for each choice, whenever a choice condition is met, the counter value is updated. The index of the maximum counter is updated continuously. The index numbers of the choices E flat, E half flat and E natural are 0, 1 and 2, respectively. As appears in the figure, the maximum count at the snapshot time was 43 and the index was 0, indicating that the identifying E is flat. Whenever a rest occurs, all counters are reset to 0, except the counter of the winning choice which is reset to 4, *i.e.* 200 milliseconds. The reason is to keep the current winner ahead of other choices by one minimum note duration (MND), approximated roughly to 200 milliseconds. This means that the output of this part will not change after a rest until another choice is detected for duration longer than MND.

The winner index numbers of both parts (instant tonic and identifying E) are cascaded to form a two-digit index to the instant maqam as illustrated in **Table 3**. In the case of our illustrating snapshots, the two-digit index was (01), indicating that the kurd is the instant maqam.

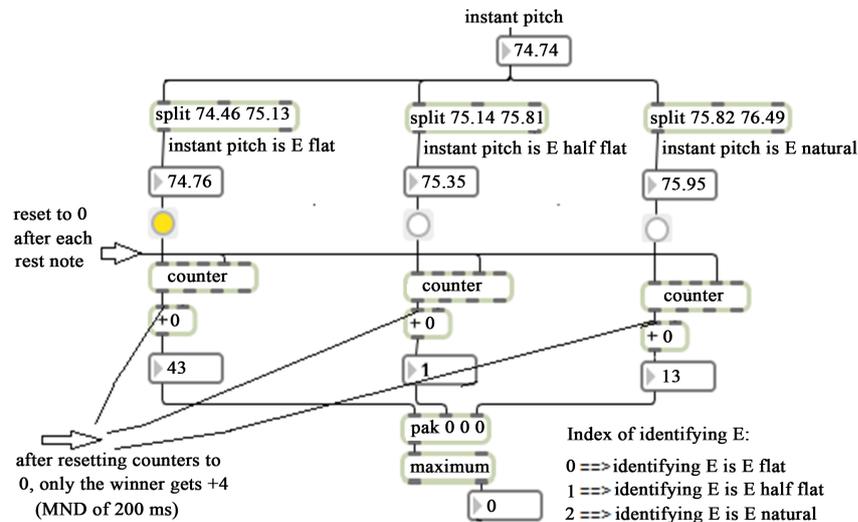


Figure 5. Max/MSP Patch fragment used to recognize the identifying E.

Table 3. Decoding two-digit indices to instant maqamat.

Two-digit index (identifying E + instant tonic)	Instant maqam
00	Nahawand
01	Kurd
10	Rast
11	Bayati
12	Sikah
20	Ajam (on C)

4.3. Competition for Long-Term Maqam

Figure 6 depicts the patch fragment used to estimate the long-term maqam. The input to this part is the index of the instant maqam, and the output is the index of the long-term maqam, *i.e.* the maqam that dominated the performance for a relatively long, yet fixed, time period. We make a competition to see which maqam index gains the maximum count over this fixed duration. Note that we replaced the two-digit indices presented above by new one-digit indices. The index numbers of the maqamat: nahawand, kurd, rast, bayati, sikah and ajam (on C) are 0, 1, 2, 3, 4 and 5, respectively. There is a counter for each choice (maqam), whenever a new instant maqam index is fed to this part; the counter belonging to this maqam is updated. The index of the winner choice is updated continuously, but is triggered only once every 4 seconds, *i.e.*, the patch finds the most performed maqam during the last 5 seconds. This duration, 4 seconds, is only a suggestion, and the user may modify it.

Every 4 seconds, all counters are reset to 0, except the counter of the winning choice which is reset to 8, *i.e.* 400 milliseconds. This is to keep the winner ahead of other choices by two MNDs, approximated roughly to 400 milliseconds. Accordingly, the output of this part will not change after the reset until another choice is detected for duration longer than two MNDs.

5. Evaluation and Discussion

This section describes the evaluation sample, also it presents and discusses evaluation results.

5.1. Evaluation Sample

Six improvisations were used for the evaluation of the proposed maqam finding model, one improvisation on each of the studied six maqamat. The durations of the improvisations ranges between 70 and 95 seconds. Since modulation is a complex issue and in order to limit the scope of this study, the improviser was requested not to modulate within the one improvisation. However, expressive performance and chromatic coloration were allowed. Quantitatively, the evaluation sample was not big, but qualitatively, we believe it was sufficient to give a clear indication on the performance of the maqam estimation model.

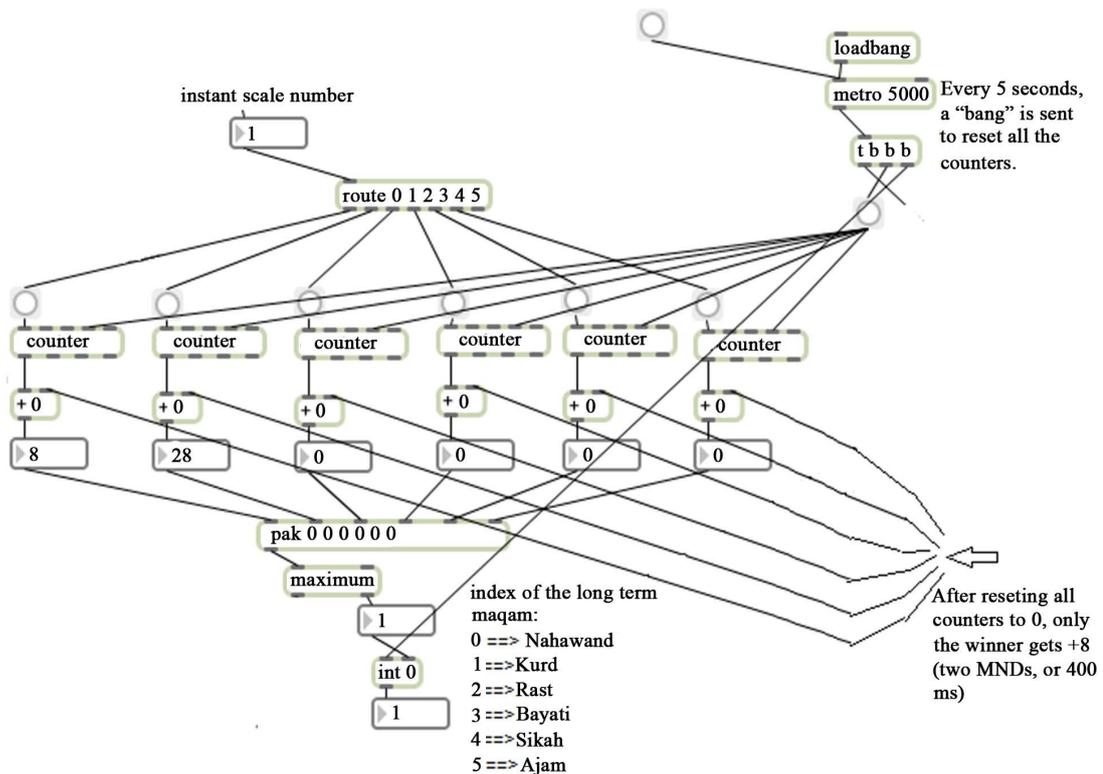


Figure 6. Max/MSP Patch fragment used to estimate the long-term maqam.

The improvisations were performed on the main nāy instrument, the dokah, by a well-experienced nāyist who has been performing in Jordan and abroad for about twenty years. In addition, the performer holds a PhD degree in education and teaches this instrument in Yarmouk University in Jordan.

5.2. Result and Discussion

Freedom of performance is a basic feature in Arab music improvisation. The instrumentalist used this freedom to express his feelings and virtuosity, as well as to show the capabilities of his/her instrument. The instrumentalist moved through some passing notes, or melodic chords (tri-, tetra- and penta-chords) that are neither the lower nor the higher chords of the improvisation maqam. This was an interesting challenge for maqam estimation, and formed a good environment to test the capabilities of the model. The rest of this section presents the evaluation results as illustrated in figures and tables, together with a discussion to the obtained results.

5.2.1. Real-Time Performance

Figure 7 and **Table 4** presented the models real-time performance. All illustrations showed the model's ability to find the maqam, as well as monitoring the instrumentalist quick passage over different melodic chords, whether those of the maqam of improvisations or others. Further discussion is provided in the following lines:

- Rast improvisation

We note from **Figure 7(a)** and the audio file (a) that instrumentalist elaborated in improvising in the maqamrastat the beginning of improvisation, while holding on his improvisation on the lower melodic tetra-chord of

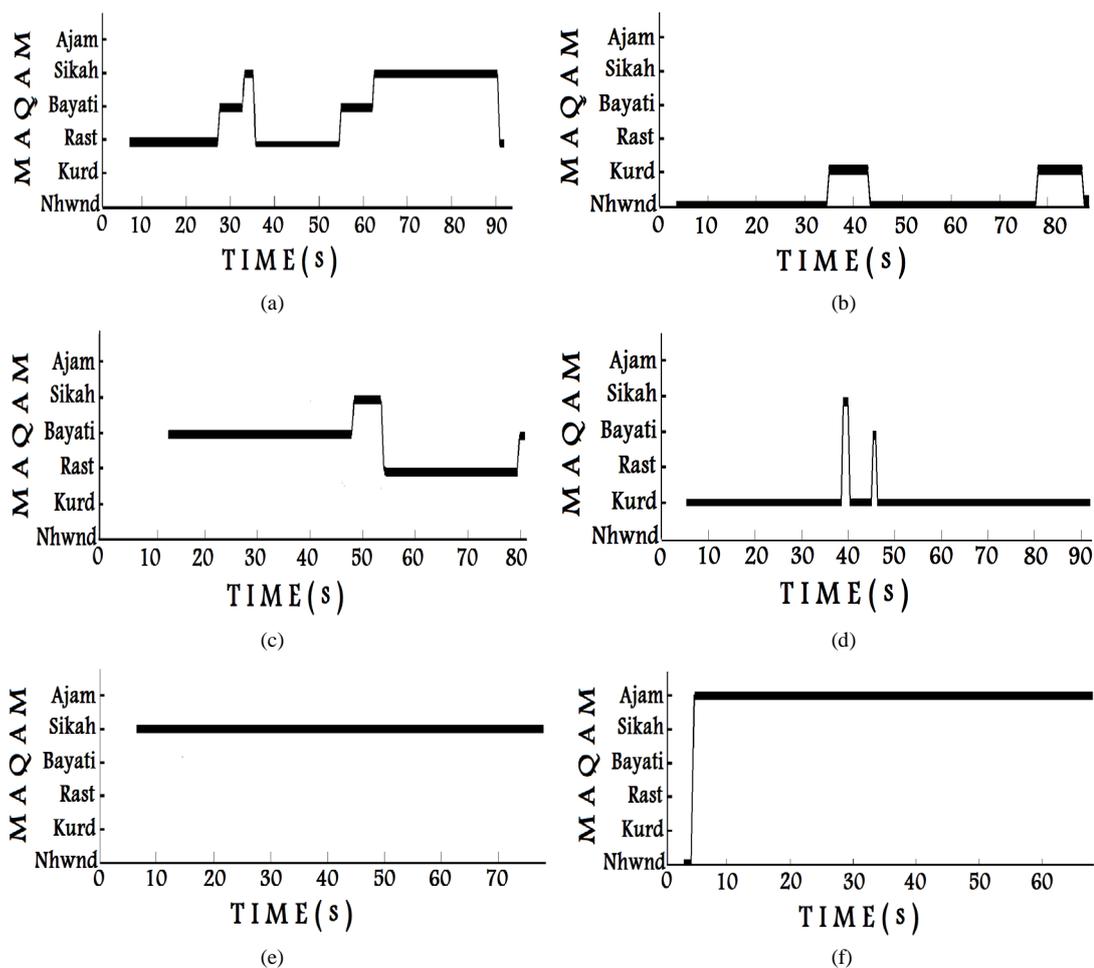


Figure 7. Real-time maqam estimation results when improvising on each of the six considered maqamat. (a) rast; (b) naha-wand; (c) bayati; (d) kurd; (e) sikah; (f) ajam (on C).

Table 4. Real-time maqam estimation results when improvising on each of the six considered maqamat (percentage of the overall duration).

Taqaṣim Maqam	Estimated Maqam (percentage of the overall duration)						
	Nahawand	Kurd	Rast	Bayati	Sikah	Ajam	Not recognized*
Nahawand	78.15%	20.81%	0%	0%	0%	0%	1.04%
Kurd	0%	94.02%	0%	1.09%	1.63%	0%	3.26%
Rast	0%	0%	45.56%	14.44%	34.44%	0%	5.56%
Bayati	0%	0%	29.94%	49.68%	7.64%	0%	12.74%
Sikah	0%	0%	0%	0%	91.45%	0%	8.55%
Ajam	1.50%	0%	0%	0%	0%	96.99%	1.50%

*Time passed before presenting the first maqamestimation result, excluding the time for initial processing (pitch detection, moving median, etc.).

this maqam. This chord has the note C as its tonic, which is also the tonic of the maqamrast. The program detected this elaboration successfully.

The instrumentalist naturally passes by the rast scale-notes D and E half-flat and makes short resolutions on these two notes. The program monitored these quick resolutions and considered them as quick coloration transpositions to the maqamatbayati and sikah. The model then detected the maqamrast again as the instrumentalist made a resolution again on the tonic, C.

After that, the instrumentalist left the lower tetrachord of the maqamrast heading to the upper notes. Therefore, the model's monitoring was generally linked to the instrumentalist's return and resolution to notes of the first tetrachord. This occurred when the instrumentalist made a resolution temporarily on the note E half-flat, and the program monitored this short resolution on E half-flat, the tonic of the maqamSika. At the end of the improvisation, the program succeeded in detecting the final cadence of the improvisation on the rast's tonic, C.

Accordingly, the performance of the model was very good as the maqamrast was the most monitored maqam in the performance, and the rast cadence was detected correctly. The model succeeded also in monitoring quick resolutions on tonics of neighboring melodic chords.

- Nahawand improvisation

We notice from **Figure 7(b)** And the audio file (b) that the model monitored the maqamnahawand successfully during 78% of the performance, including the final cadence. This correlates with the several short cadences throughout the performance as heard in the audio file. Most of the cadences were on the nahawand's tonic, C, and only few short cadences were on the note D—the tonic of the maqamkurd. This is normal as bothmaqamat share the same tones in their lower parts. The model did not detect any of the maqamat having the neutral note E half-flat or the natural note E as scale notes. This is also an indication to the robustness of the model because neither the intervals of the maqamnahawand nor the audio performance includes such notes.

- Bayati improvisation

The audio file (c) and **Figure 7(c)** both showed the instrumentalist's elaborated performance on the maqam-bayati in the first part of the improvisation, and this is a conventional approach of improvisation. Afterwards, the instrumentalist added some coloration to the improvisation by introducing short resolutions on other scale-tones than the tonic, D. The figure indicates short resolutions on Mid and C; the tonics of the sikah and rastmaqamat, respectively. The model also detected the final cadence of the improvisation successfully.

- Kurd improvisation

Both **Figure 7(d)** and the audio file (d) shows that the instrumentalist elaborated in the maqamkurd throughout the improvisation without adding coloration transpositions. This is why the model detected the maqamkurd almost in all the way through the end of the improvisation.

The figure also showed two sparks indicating very short rests on other maqamat. This returned to the elaborated ornamentation and articulation performed by the instrumentalist. As could be heard from the wave file, the instrumentalist used vibrato, trill, tremolo and combinations of these throughout the performance. However, these sparks, together, forms less than 3% of the total detection results. So this did not change the fact that the model was accurate in detecting the maqamkurd almost throughout the improvisation. The model also succeeded in detecting the final cadence of on the kurd's tonic, D.

- Sikah improvisation

Both **Figure 7(e)** and the audio file (e) shows that the instrumentalist elaborated in the maqamsikah throughout the improvisation. The instrumentalist kept moving among the maqamsikah tones, but he tends not to resolute but on the maqam’s tonic, Ed.

In order to express the distinguished feeling of the maqamsikah, the instrumentalist did not resolute on any of the two tones below the tonic, neither D nor C. This is to avoid the different feelings of the maqamatbayati and rast, respectively. The final cadence on the tonic of sikah was also detected successfully.

- Ajam improvisation

We notice from **Figure 7(f)** And the audio file (f) that the instrumentalist started his performance by a slight coloration in intonation, he nearly approached the tone E flat before letting the listener realizes the strong feeling of ajam on C assured by the tone E. The performance on ajam, then, continued throughout the improvisation, and the model monitored that successfully. The final cadence of the improvisation on the maqam’s tonic, C, was also detected successfully.

5.2.2. Five-Second Buffer Performance

Figure 8 and **Table 5** presented the models performance when the audio buffer is five second. Illustrations showed an improved performance for the long buffer when compared to the real-time settings of the model. The model was granted more time to detect and monitor the maqam. This made the model informed of a larger number of tones and melodic progressions, and this made the features of the maqam clearer.

When comparing the **Table 4** and **Table 5**, we note that the percentages of detecting the performance maqam in the overall duration are better in this experiment than the previous one, especially for the maqamatrast and bayati. These percentages are presented more clearly in **Table 6** under the name “percentage of confidence”. It

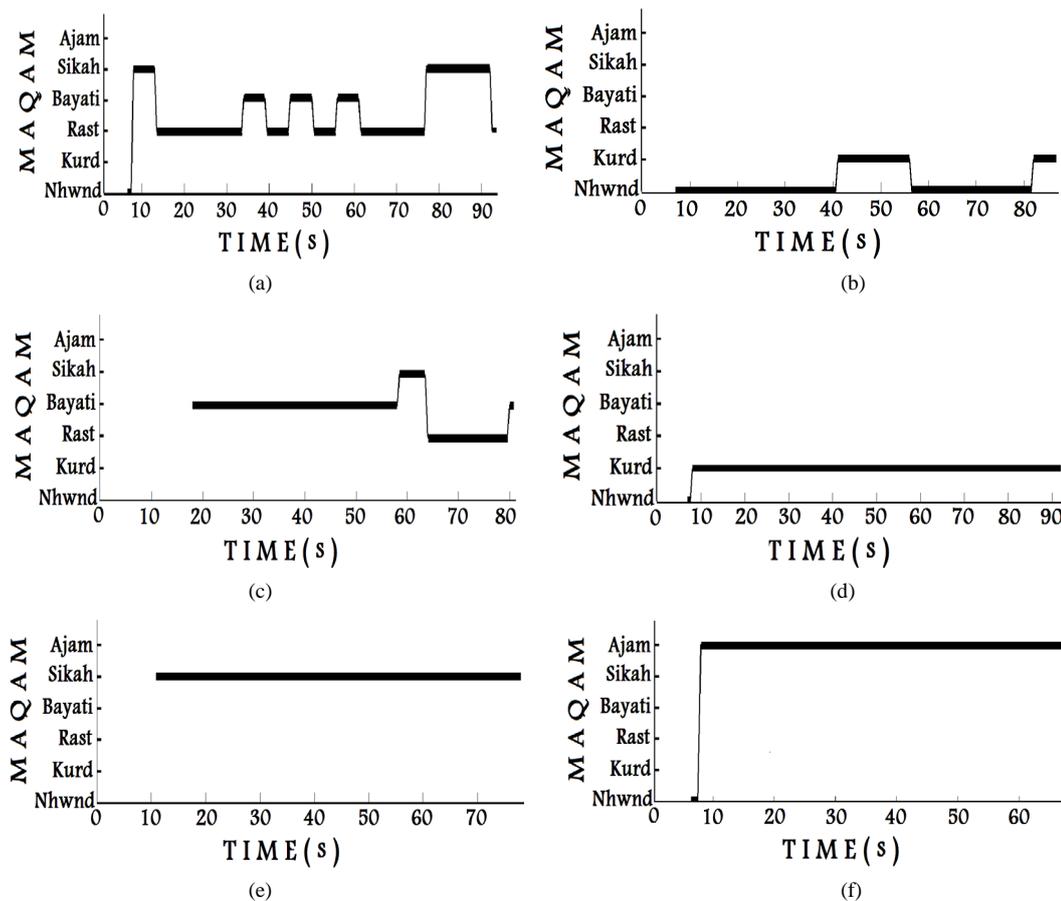


Figure 8. Five-second-buffer maqam estimation results when improvising on each of the six considered maqamat. (a) rast; (b) nahawand; (c) bayati; (d) kurd; (e) sikah; (f) ajam (on C).

Table 5. Five-second-buffer maqam estimation results when improvising on each of the six considered maqamat (percentage of the overall duration).

Taqaşim Maqam	Estimated Maqam (percentage of the overall duration)						
	Nahawand	Kurd	Rast	Bayati	Sikah	Ajam	Not recognized [*]
Nahawand	73.75%	26.25%	0%	0%	0%	0%	0%
Kurd	0.59%	99.41%	0%	0%	0%	0%	0%
Rast	0.58%	0%	56.07%	19.08%	24.28%	0%	0%
Bayati	0%	0%	21.62%	56.76%	7.43%	0%	14.19%
Sikah	0%	0%	0%	0%	95.07%	0%	4.93%
Ajam	0.82%	0%	0%	0%	0%	99.18%	0%

^{*}Time passed before presenting the first maqam estimation result, excluding the time for initial processing (pitch detection, moving median, etc.) and the 5-second buffer.

Table 6. Percentages of confidence.

Taqaşim Maqam	Confidence [*] (Real-time)	Confidence ^{**} (Five-second-buffer)
Nahawand	78.15%	73.75%
Kurd	94.02%	99.41%
Rast	45.56%	56.07%
Bayati	49.68%	56.76%
Sikah	91.45%	95.07%
Ajam	96.99%	99.18%
Average	75.98%	80.04%
Standard Deviation	22.93%	20.61%

^{*}At any time during performance, the percentage to which the instantly detected maqam can be true. ^{**}At any time during performance, the percentage to which the detected maqam can be true.

is the identified as the percentage to which the detected maqam (in real-time or in 5-second buffer) can be the same maqam of the instrumental performance. This is provided that we can check the detection result at any time during the performance.

Table 6 shows that the average of percentage of confidence in this experiment is higher than that of the previous experiment by 4.06%. The standard deviation of results of the different maqamat is less by 2.32%. This indicates that percentages of confidence are less scattered in the new experiment, and this is a good indication. It was also remarkable that the expansion of the buffer size helped in eliminating misleading sparks caused by elaborated ornamentations and articulations. This is illustrated clearly in the kurd figures; **Figure 7(d)** showing the results with a short buffer size and **Figure 8(d)** showing the results with a long buffer size. The later had no mis-guiding sparks.

On the other hand, expanding the buffer size had two side effects. The first is expanding the time passed before presenting the first maqam estimation result. The second is that the final cadence of the maqamahawand was not detected correctly, see **Figure 8(b)**. This is because the duration of the tonic cadence was very short when compared to the long buffer size. However, the general detection result of this maqam as expressed by the term “confidence” was very good, 73.75%, see **Table 6**. In addition, the cadence was detected correctly in all other maqamat, see **Figure 8**.

5.2.3. Full-Duration Performance

The Full-duration performance is described as the ability of the model to estimate the right maqam after the completion of the full improvisation. The maqam having the highest percentage of the overall duration is considered

the final estimation results. Accordingly and as shown in **Table 4** and **Table 5** the model succeeded in estimating all the improvisation maqamat because the maqam of improvisation always had the highest percentage of the overall duration whether in the real-time or the five-second buffer experiments.

6. Conclusions and Future Work

We presented a real-time maqam estimation model configured for the nāydukah. The model detected the tonic and the identifying E (E, E half-flat and E flat) of each melodic figure, and used them to predict the maqam in real-time. Accumulated prediction results were used to estimate the maqam every five seconds and also over the full duration. Six improvisations on six different maqamat were used in the evaluation. The model estimated all the maqamat correctly over the full duration; real-time and five-second estimation results were ambitious with average confidence of 75.98% and 80.04%, respectively. The very good performance of the model in monitoring melodic progressions makes it suitable for useful applications of admirable implication on education as well as on multimedia live performances and accompaniment. Future work can include expanding the idea of the “identifying tone” by considering several tones. This is in order to allow for the estimation of several other maqamat in different sound ranges.

References

- [1] Al-Taee, M., Al-Rawi, M. and AL-Ghawanmeh, F. (2008) Time-Frequency Analysis of the Arabian Flute (Nay) Tone Applied to Automatic Music Transcription. *Proceedings of the 6th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-08)*, Doha, 31 March-4 April 2008, 891-894. <http://dx.doi.org/10.1109/aiccsa.2008.4493636>
- [2] Tayseer, A., Haddad, R., Sukkarieh, H. and Al-Ghawanmeh, F. (2014) Al-maqamāt fi al-oghniyah Al-Arabia al mo'asirah [Arab Maqamat in Contemporary Arab Songs]. *Dirasat Journal of Human Sciences*, **41**, 598-611.
- [3] Al-Ghawanmeh, F. (2013) Automatic Accompaniment to Arab Vocal Improvisation “Mawwāl”. Unpublished Master’s Thesis, New York University, New York City.
- [4] Haddad, R., AL-Ghawanmeh, F. and AL-Ghawanmeh, M. (2010) Educational Tools Based on MIR System for Arabian Woodwinds. *Journal of Music, Technology and Education*, **3**, 31-45. http://dx.doi.org/10.1386/jmte.3.1.31_1
- [5] Fahmi, A. (2005) Utilization of Common Melodies in Teaching Nāy to Beginners. Internal Report, Higher Conservatory for Arabian Music, Cairo.
- [6] Al-Gubbi, H. (2004) The Acoustics and Arabian Music in View of Informatics Tools. *Journal of Music Research*, **3**, 15-32. Arab Academy of Music—Arab League.
- [7] Al-Taee, M., Al-Ghawanmeh, M., Al-Ghawanmeh F. and Omar, B. (2009) Analysis and Pattern Recognition of Arabian Woodwind Musical Tones Applied to Query-By-Playing Information Retrieval. *Proceedings of the International Conference of Computer Science and Engineering, ICCSE_59*, World Congress on Engineering, WCE. London, U.K., 1-3 July.
- [8] Al-Taee, M. and Al-Ghawanmeh, F. (2010) Tahlīlwaistikhlās al-khasa'is al-mūsiqīyahlinaghāmālat al-nāyAlArabi [Analysis and Features Recognition of Arabian Flute (Nay) Musical Tones], *Jordan Journal of the Arts*. Yarmouk University, Irbid.
- [9] Al-Ghawanmeh, F., Jafar, I., Al-Taee, M., Al-Ghawanmeh, M. and Muhsin, Z. (2011) Development of Improved Automatic Music Transcription System for the Arabian Flute (NAY). *Proceedings of the 8th International Multi-Conference on Systems, Signals and Devices (SSD-11)*, Sousse, 22-25 March 2011, 1-6. <http://dx.doi.org/10.1109/SSD.2011.5993561>
- [10] AL-Ghawanmeh, F., AL-Ghawanmeh, M. and Haddad, R. (2009) Appliance of Music Information Retrieval System for Arabian Woodwinds in E-Learning and Music Education. *Proceedings of the International Computer Music Conference, ICMC*, Montréal, 16-21 August 2009, 17-20. Ann Arbor, MI: MPublishing, University of Michigan Library.
- [11] Bedair, R. (n.d.). Abhath al-tatwir fi alat al-naiallatiqamabiha RedaBedair [Researches on Improving the Nay Instrument by the NayistRedaBedair]. <http://www.redabedair.com/tgarob.asp>
- [12] Al-Ghawanmeh, F., Al-Ghawanmeh, M. and Obeidat, N. (2014) Toward an Improved Automatic Accompaniment to Arab Vocal Improvisation, Mawwāl. *Proceedings of the 9th Conference on Interdisciplinary Musicology (CIM14)*, Berlin, 4-6 December 2014, 397-400.
- [13] AL-Ghawanmeh, F., Haddad, R. and AL-Ghawanmeh, M. (2014) Proposing a Process for Using Music Analysis Software to Improve Teaching Authentic Arab Singing and Ornamenting. *International Journal of Humanities and Social Science*, **4**, 47-56. Los Angeles (USA): Centre for Promoting Ideas.

- [14] Mawaweel: Web Application for Automatic Accompaniment to Arab Vocal Improvisation ‘Mawwāl’. (2013). <http://www.mawaweel.com>
- [15] Al-Ghawanmeh, F. and Shannak, Z. (2014) Automatic Accompaniment to Arab Vocal Improvisation: From the Technical to the Commercial Perspectives. *Journal of Software Engineering and Applications*, **8**, 16-25. Delaware (USA): Scientific Research Publishing.
- [16] Zhu, Y. and Kankanhalli, M. (2003) Music Scale Modeling for Melody Matching. *Proceedings of the Eleventh ACM International Conference on Multimedia*, 2-8 November 2003, Berkeley, 359-362. <http://dx.doi.org/10.1145/957013.957091>
- [17] Kolinski, M. (2015) Mode. Encyclopædia Britannica. <http://www.britannica.com/EBchecked/topic/386980/mode>
- [18] The Arabic Maqam (2007) In Maqam World. <http://www.maqamworld.com/maqamat.html>
- [19] Izmirli, O. (2005) Template Based Key Finding from Audio. *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, 211-214.
- [20] Zhu, Y., Kankanhalli, M. and Gao, S. (2005) Music Key Detection for Musical Audio. *Proceedings of the 11th International Multimedia Modelling Conference*, Melbourne, 12-14 January 2005, 30-37. <http://dx.doi.org/10.1109/MMMC.2005.56>
- [21] Peeters, G. (2006) Chroma-Based Estimation of Musical Key from Audio-Signal Analysis. *Proceedings of the International Symposium for Music Information Retrieval (ISMIR)*, Victoria, CA, 115-120.
- [22] Bello, J. (2011) Characterizing Harmony from Audio. Teaching Material. New York University, New York.
- [23] Peeters, G. (2006) Musical Key Estimation of Audio Signal Based on Hidden Markov Modeling of Chroma Vectors. *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Montreal, 18-20 September 2006, 127-131.
- [24] Gedik, A.C. and Bozkurt, B. (2008) Automatic Classification of Turkish Traditional Art Music Recordings by Arel Theory. *Proceedings of the 4th Conference on Interdisciplinary Musicology (CIM08)*, Thessaloniki, 3-6 July 2008, 10 p.
- [25] Ünal, E., Bozkurt, B. and Karaosmanoğlu, M. (2012) Incorporating Features of Distribution and Progression for Automatic Makam Classification. *2nd CompMusic Workshop*, Bahçeşehir Üniversitesi, Istanbul, July 2012.
- [26] Chuan, C.H. and Chew, E. (2005) Polyphonic Audio Key Finding Using the Spiral Array CEG Algorithm. *IEEE International Conference on Multimedia and Expo*, 6-8 July 2005, 21-24). <http://dx.doi.org/10.1109/ICME.2005.1521350>
- [27] Cycling’74 Website. <https://cycling74.com/>
- [28] Hove, M.J., Sutherland, M.E. and Krumhansl, C.L. (2010) Ethnicity Effects in Relative Pitch. *Psychonomic Bulletin & Review*, **17**, 310-316. <http://dx.doi.org/10.3758/PBR.17.3.310>
- [29] Robertson, A. and Plumbley, M. (2007) B-Keeper: A Beat-Tracker for Live Performance. *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, New York, 234-237. <http://dx.doi.org/10.1145/1279740.1279787>
- [30] Puckette, M., Apel, T. and Zicarelli, D. (1998) Real-Time Audio Analysis Tools for PD and MSP. *Proceedings of the International Computer Music Conference*, Michigan, 109-112.
- [31] Apel, T. (2015) Max and Max for Live Patches and Externals. <http://vud.org/max/>
- [32] Yoo, L. and Fujinaga, I. (1999) A Comparative Latency Study of Hardware and Software Pitch-Trackers. *Proceedings of the International Computer Music Conference*, Beijing, 36-40. <http://gigue.peabody.jhu.edu/~ich/research/icmc99/icmc99.pitch.pdf>
- [33] Al-Sho’lah, M. and Jamal, A. (2009) al-diwan al-’am fi sharh al-maqam [Explaining the maqam]. Dar Al-Kitab Al-Hadith, Cairo.
- [34] Arabic Maqam Index (2007) In Maqam World. <http://www.maqamworld.com/maqamindex.html>
- [35] Gedik, A.C. and Bozkurt, B. (2010) Pitch-Frequency Histogram-Based Music Information Retrieval for Turkish Music. *Signal Processing*, **90**, 1049-1063. <http://dx.doi.org/10.1016/j.sigpro.2009.06.017>
- [36] Abbas, H. (1986) Nazariyyat al-musiqā al-arabiya [Arab Music Theories]. Al-Hurriyah Press, Baghdad.

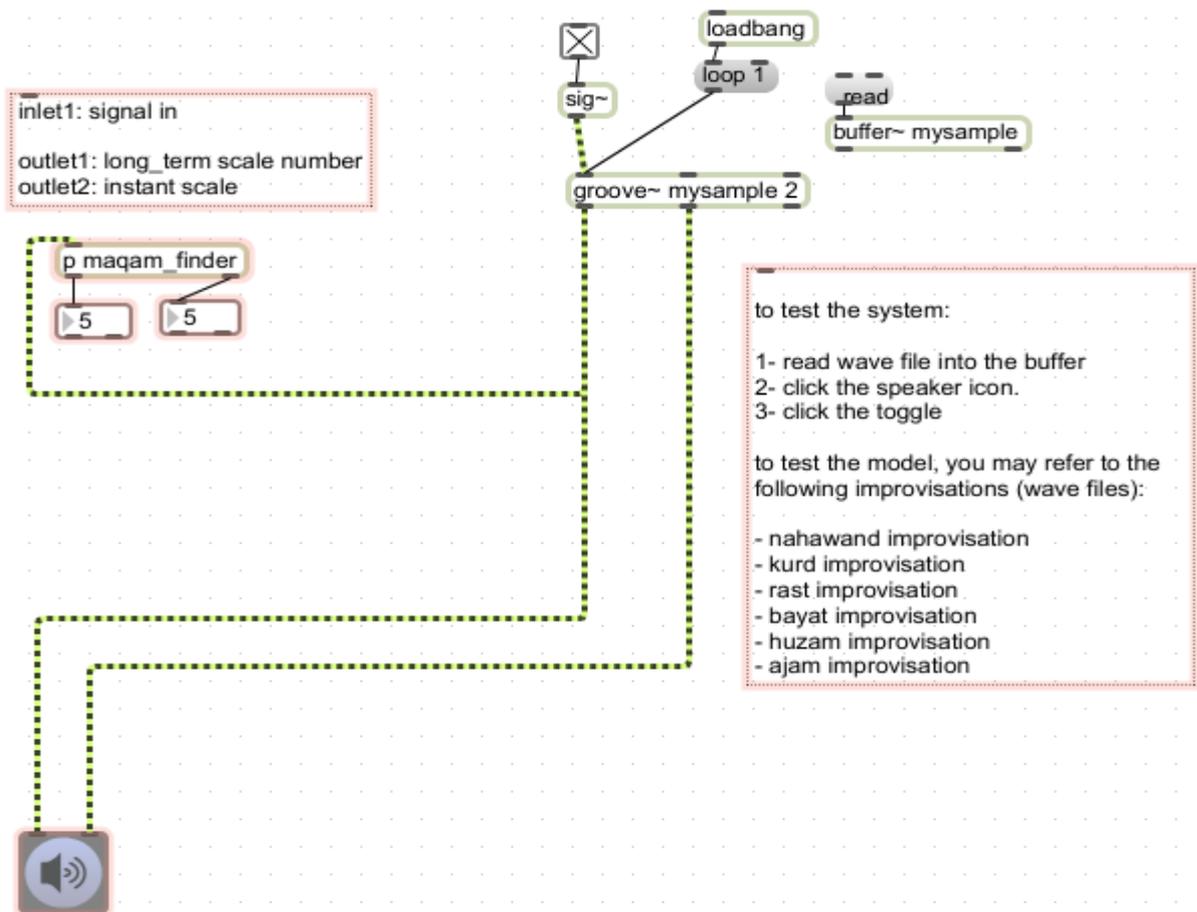
Appendices

Appendix 1: Audio Files

- A. Audio file (a) is for nāy improvisations on maqamrast
 - B. Audio file (b) is fornāy improvisations on maqamnahawand
 - C. Audio file (c) is for nāy improvisations on maqambayati
 - D. Audio file (d) is fornāy improvisations on maqamkurd
 - E. Audio file (e) is for nāy improvisations on maqamsikah
 - F. Audio file (f) is fornāy improvisations on maqamajam on C
- Please contact the principal investigator to access the audio files.

Appendix 2: Useful Figures

- A. Abstracted overview of the model



B. Panorama view of the model

