Automatic Music Information Retrieval Using a Microphone Array

Kerem Okyay
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
kerem.okyay@student.kuleuven.be

Thant Maung
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
thant.maung@student.kuleuven.be

Alken Rrokaj
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
alken.rrokaj@student.kuleuven.be

Fatjon Barçi
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
fatjon.barçi@student.kuleuven.be

Wannes Verstraeten
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
wannes.verstraeten@student.kuleuven.be

***Abstract*—In this paper, we explore the field of music information retrieval using a microphone array in the context of music therapy. The goal is to reduce the time required for therapists to accurately diagnose autism spectrum disorder (ASD) and enhance the effectiveness of music therapy treatment for these individuals. Our approach combines blind signal source separation, onset detection and automatic music transcription techniques. We evaluate our approach on music therapy session recordings. Although the results of this work were not optimal, we discuss the reasons for this and suggest directions for future research.**

***Index Terms*—music therapy, automatic music transcription, onset detection, source separation**

I. Introduction

In this paper, we will be addressing the problem of music information retrieval (MIR) in the context of music therapy. Music therapy is a form of treatment that utilizes music and musical elements, as well as instrumental improvisation, to facilitate communication, relationships, and expression. It is typically led by a trained therapist and conducted in a group setting with patients. In our case, the group of patients consists of children with ASD.

We aim to use signal processing tools to automate the analysis of these sessions, with the ultimate goal of reducing the time it takes for therapists to classify a child as having ASD or not, and to help them increase the effectiveness of the treatment.

We begin by providing an overview of related works and state-of-the-art methods in the field. We then proceed to discuss the three main stages of our proposed process: blind source separation, onset detection and automatic music transcription. We explore approaches including principal component analysis (PCA), independent component analysis (ICA), NINOS^2 algorithm and Non-Negative Matrix Factorization (NMF).

II. Related Work

A. Related work: blind source separation

Source separation in general is the process of separating and extracting individual sound sources or elements within an audio mixture. The goal of source separation is to extract the individual sources in a way that allows them to be analyzed, modified, or re-combined in new ways. It is a widely used technique in fields such as music production, speech processing, and audio signal processing, and is an active area of research in the field of signal processing.

Blind Source Separation (BSS) is the process of extracting individual source signals from a mixture of signals without the prior knowledge of the source positioning or mixing system.

1) ICA: ICA achieves BSS by finding nonlinear, time-delayed, or non-stationary decorrelation in the mixed signals, which allows for the separation of the source signals. It is a purely statistical process, the separation mechanism has not been clearly understood in the sense of acoustic signal processing, and it has been difficult to know which components were separated, and to what degree. [1]

There exist linear and non-linear implementations for ICA. The method described in this paper works using the linear implementation as the non-linear model is more complex to implement, is sensitive to initialization which can affect the final solution. However, a recent publication [2] shows that a newly developed non-linear model can now not only outperform the existing models in terms of accuracy, but also speeds up the convergence of blind source separation.

B. Related work: note onset

Note onset can techniques can be dividend into two broad categories. The first methods in NOD are the non-data driven (or traditional) approaches. These non-data driven methods can be further subdivided into probabilistic methods and non-probabilistic methods [3]. Some of the more recent non-probabilistic non-data driven methods include LSF [4] and

NINOS^2 [5].

The data-driven approaches are mainly focused on machine learning techniques. The best results come from convolutional neural networks (CNN) [6]. These CCNs gained popularity through their use in image processing, and since spectrograms can easily be regarded as images, this was a logical next step from the first attempts using only feed-forward neural networks [7]. Other machine learning methods have also been used, e.g. recurrent neural networks have proven especially interesting in real time applications [8].

C. Related work: non-negative matrix factorization

Non-negative matrix factorization (NMF) has been widely used in the field of automatic music transcription (AMT) due to its ability to decompose a matrix of spectrogram data into interpretable basis and coefficient matrices, where the basis matrix represents the spectral templates of musical components and the coefficient matrix represents their temporal activations.

One early application of NMF was presented by Lee and Seung in [9], where they demonstrated that NMF can effectively learn parts of a non-negative matrix and decompose it. This approach was then applied to polyphonic music by Smaragdis and Brown [10] and it has been shown that the method had promising results. The NMF model has been modified in various ways to incorporate additional structure in the dictionary and the activations. One example of this is the use of sparsity in the activations, which leads to a solution with only a few, but significant, activations. This approach, known as sparse coding, has been successful and has led to the development of various methods in which the size of the dictionary can be much larger than the input dimension [11]. Other extensions of NMF involve designing the dictionary in a supervised manner, using additional training material. For instance, given K recordings that each contain a single note, the dictionary can be pre-computed and fixed e.g., each piano note recorded individually, K=88. This way, the templates are guaranteed to be free of interference from other notes and also to have a clear interpretation.

Overall, NMF has been widely used in the field of AMT. In this paper we reuse the unsupervised version of NMF discussed in this paper [10] and add an extra step in the beginning using an onset detection algorithm to estimate its reduced rank.

III. Blind Source Separation

A. Principle Component Analysis

PCA is a statistical technique that is used to reduce the number of dimensions in a dataset while retaining as much information as possible. It does this by finding the directions in which the data varies the most, and projecting the data onto a lower-dimensional space along these directions. The dimensions of the lower-dimensional space are called the "principal components", and they are chosen to be orthogonal (i.e., perpendicular) to one another. PCA was first formulated by Karl Pearson in 1901 as "The linear projection that minimizes the average projection cost, defined as the mean squared distance between data points and their properties" [12]. An alternative and more intuitive definition was independently formulated by Harold Hoteling in 1936: "The orthogonal projection of the data onto a lower dimensional linear space, known as the principle subspace, such that the variance of the projected data is maximized" [13].

1) Finding the components: In PCA, the principal components are obtained by performing singular value decomposition (SVD) on the data matrix X. SVD decomposes the matrix X as a product of three matrices, more specifically

$$X = P \cdot \triangle \cdot \mathbf{Q}^{\mathrm{T}} \tag{1}$$

where P is the matrix of left singular vectors, Δ is the diagonal matrix of singular values, and Q is the matrix of right singular vectors, then the principal components are represented by the columns of the matrix Q. The matrix Q is also known as the loading matrix, and it represents the direction cosines of the principal components.

The matrix X can be written as the product of the factor scores F and the transpose of the loading matrix

$$X = FQ^{\mathrm{T}} \tag{2}$$

This representation can be useful for identifying patterns or trends in the data. The principal components can also be visualized geometrically by rotating the original axes, using the matrix Q as the matrix of direction cosines.

B. Individual Component Analysis

ICA is a signal processing technique used to separate a multi-dimensional signal into its independent components. ICA is motivated by the central limit theorem, which states that the sum of a large number of independent, identically distributed random variables will tend to be distributed normally. Therefore, by finding a linear transformation that results in components that are as independent as possible, ICA aims to separate out the independent components of a signal and isolate their contributions.

ICA is a blind source separation technique, meaning that it can separate out components of a signal without any prior knowledge of the source or the signal. The main idea behind ICA is to find a linear transformation of the observed signals, such that the components of the transformed signals are statistically independent. One way to achieve this is by maximizing the non-Gaussianity of the transformed signals.

Kurtosis is a measure of non-Gaussianity, and is often used to quantify the degree of deviation from a Gaussian distribution. It is calculated using the following formula:

$$K = \frac{E[(y - \overline{y})^4]}{(E[(y - \overline{y})^2])^2} - 3 \tag{3}$$

where $\overline{y}$ is the sample mean of the extracted signals y. The constant 3 is included to ensure that Gaussian signals have a kurtosis of zero, while super-Gaussian signals have

a positive kurtosis and sub-Gaussian signals have a negative kurtosis. The denominator, which represents the variance of y, is included to ensure that the measured kurtosis takes into account the variance of the signal.

By maximizing the kurtosis of the transformed signals, ICA aims to find a linear transformation that results in components that are as non-Gaussian as possible. This helps to ensure that the transformed components are as independent as possible, and enables ICA to separate out the independent components of the original signal.

C. Implementation

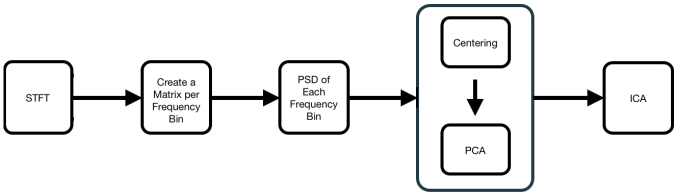

Fig. 1. Source Separation Algorithm Flowchart

The steps involved in our blind source separation algorithm using ICA are outlined in the flowchart shown in Figure 1. These steps are designed to perform source separation on a set of recordings and obtain estimates of the independent sources. The specific steps are as follows:

1) Convert all the recordings in the frequency domain using the Short-Time Fourier transform (STFT).
2) Create a matrix for each frequency bin with rows being the result from each measurement.
3) Calculate the power spectral density (PSD) for each frequency bin.
4) Center the data by subtracting the mean from each measurement.
5) Reduce the dimensionality of the data using PCA to the number of expected sources.
6) Separate the sources using ICA.
7) Convert the results back into complex form to prepare them for the inverse short-time Fourier transform (ISTFT).
8) Run the separated sources through the ISTFT to obtain the final separated signals in the time domain.

D. Methods for Evaluating ICA Performance

There are several methods that can be used to evaluate the performance of ICA algorithms for source separation. Here, we describe a few of the most commonly used methods:

1) Reconstruction Error: One of the most straightforward ways to evaluate the performance of an ICA algorithm is to compare the original mixed signals to the reconstructed signals obtained by applying the ICA algorithm and then mixing the resulting independent components back together. The reconstruction error is then calculated as the mean squared error (MSE) between the original and reconstructed signals.

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2 \tag{4}$$

where x_t is the original signal at time t, $\hat{x}_t$ is the reconstructed signal at time t, and T is the total number of time steps.

2) Source-to-Interference Ratio (SIR): The source-to-interference ratio (SIR) is a measure of the separation between the source signals in the reconstructed signals. It is defined as the ratio of the energy of the desired source signal to the energy of the interference (i.e., the sum of all other source signals).

$$\text{SIR} = 10\log_{10}\left(\frac{\sum_{t=1}^{T} s_t^2}{\sum_{t=1}^{T} i_t^2}\right) \tag{5}$$

where s_t is the desired source signal at time t and i_t is the interference at time t.

3) Signal-to-Noise Ratio (SNR): The signal-to-noise ratio (SNR) is a measure of the quality of the reconstructed signals compared to the original signals. It is defined as the ratio of the energy of the desired signal to the energy of the noise (i.e., the difference between the original and reconstructed signals).

$$\text{SNR} = 10\log_{10}\left(\frac{\sum_{t=1}^{T} s_t^2}{\sum_{t=1}^{T}(s_t - \hat{s}_t)^2}\right) \tag{6}$$

where $\hat{s}_t$ is the reconstructed signal at time t.

E. Results

Unfortunately, we were unable to obtain meaningful results from our independent component analysis (ICA) algorithm due to a lack of labeled data and some implementation mistakes in our Python script. Additionally the labeled data we had are likely to have gone through non linear mixing which is a fundamental assumption for ICA to perform as expected.

1) Lack of Labeled Data: One of the main challenges we faced was the lack of labeled data for training and evaluating our ICA algorithm. In order to effectively separate the mixed signals into their underlying independent components, it is important to have a set of reference signals that can be used to train and evaluate the algorithm. Without this labeled data, it is difficult to accurately assess the performance of the algorithm and to make any meaningful conclusions about the results.

2) Implementation Mistakes: In addition to the lack of labeled data, we also encountered some implementation mistakes in our Python script due to our lack of experience using the relevant libraries. These mistakes likely contributed to the inability to obtain meaningful results.

For example, we may have made errors in the way we converted the power spectral density back into complex form, leading to the inability to obtain meaningful results.

F. Conclusion on the method

In conclusion, the lack of labeled data,the non-linearities in the mixing and implementation mistakes in our Python script were major obstacles in our attempt to obtain meaningful results from our ICA algorithm for source separation. In future work, it will be important to address these issues in order to more accurately assess the performance of the algorithm and to make more meaningful conclusions about the results.

IV. Automatic music transcription: Note onset using NINOS^2

Automatic music transcription (AMT) is the process of converting audio of musical recordings into musical notation. This task is challenging for computers due to the complex structure of music, which can involve multiple instruments playing simultaneously (polyphonic music) or just one instrument playing at a given time (monophonic music). In the field of music signal processing AMT is still considered as a fundamental problem [14]. AMT systems typically take an audio file as input, perform time-frequency analysis, and output a representation of pitches over time including the onset, offset, and estimated fundamental frequency.

Note onset detection is the first step in our pipeline of automatic music transcription. We use the results obtained in this step both for the final onset times and as an intermediate step in the automatic music transcription, as explained in section V. The algorithm used is NINOS^2 [5].

A. Note onset

Note onset detection (NOD) is a form of event detection in music signals. The goal is to determine the exact onset time for all the notes in a piece. Precisely defining the onset of a note is difficult. In general notes are assumed to consist of three parts: the ”attack”, the ”sustain”, and the ”decay”. The attack is a generally considered as the most interesting part for NOD. It consist of a short period in which an abrupt change in signal energy occurs. The beginning of this attack is assumed to be the onset of a note.

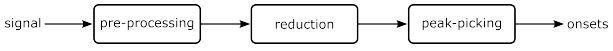

Fig. 2. Flowchart visualising the general process of NOD.

In most cases NOD is split into three steps as depicted in figure 2:

1) **Pre-processing** is an optional step but in most cases of NOD this is either time-frequency transformation and or noise reduction.
2) **Reduction** refers to the mapping of the input signal to a highly subsampled *detection function* [3]. This step is the crux of NOD.
3) **Peak-picking**

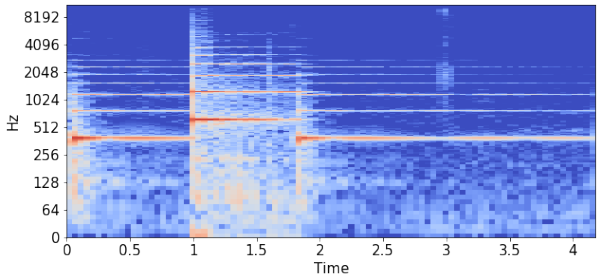

Fig. 3. STFT spectrogram of three piano notes

B. NINOS^2

The method used here is a form of inverse sparsity measure, meaning that it returns high values for vectors where the energy is spread out and low values for vectors with concentrated places of energy. Looking at a piece of music we see that most energy is concentrated in fundamentals and harmonics of the played notes. However, note onsets contain energy that is spread over a larger frequency range. This explains why we are mainly interested in the low energy components as explained later.

To calculate the note onset detection function (ODF) we start from a STFT spectrogram as shown in figure 3. From this we obtain a vector $\mathbf{y}$ for each time frame of the STFT as follows:

$$X_k(n) = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} w(m)x(n+m)e^{-\frac{2j\pi mk}{N}} \tag{7}$$

$$Y_k(n) = \log(\lambda|X_k(n)| + 1) \tag{8}$$

where λ is a compression parameter. From Y_k we finally construct $\mathbf{y}$ which is a vector of the J lowest energy bins of Y_k with

$$J = \left\lfloor \frac{\gamma}{100}\left(\frac{N}{2} - 1\right)\right\rfloor \tag{9}$$

where γ is the percentage of STFT frequency bins in the low energy subset. Using all the $\mathbf{y}$ and J we can now calculate the ODF for the entire STFT using equation (10). Resulting in a trace as shown in figure 4.

$$\boxed{\aleph_{\ell_2\ell_4}(n) = \frac{\|\mathbf{y}(n)\|_2}{\sqrt[4]{J} - 1}\left(\frac{\|\mathbf{y}(n)\|_2}{\|\mathbf{y}(n)\|_4} - 1\right)} \tag{10}$$

It is important to note that the inverse sparsity measure does not exclusively work for the L_2 and L_4 norm ratio. Other norm ratios can be used and especially the L_1, L_2 variant simplifies to a much less computationally expensive form that gives comparable results to the original.

$$\aleph_{\ell_1}(n) = \frac{\|\mathbf{y}(n)\|_2}{\sqrt{J} - 1}\left(\frac{\|\mathbf{y}(n)\|_1}{\|\mathbf{y}(n)\|_2} - 1\right) \tag{11}$$

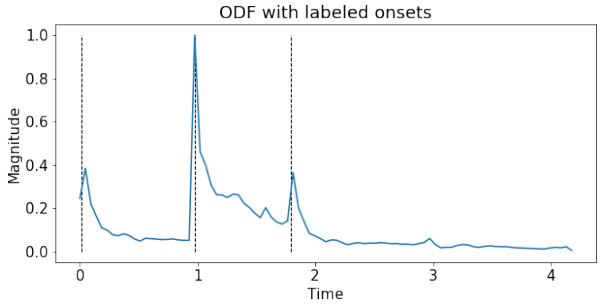

Fig. 4. NINOS[2] onset detection trace calculated on the STFT of figure 3 with the dotted lines representing the actual onsets

C. Peak picking

The peak-picking algorithm used is one commonly encountered in state-of-the-art NOD literature. We use the variant of [5], where three conditions have to be met in order for a point to be considered as an onset:

$$1)\ \aleph(i) = \max_{l} \aleph(i+l) \tag{12}$$

$$1)\ \aleph(i) \geq \frac{1}{a+b+1} \sum_{l=-1}^{b} \aleph(i+l) + \delta \tag{13}$$

$$1)\ i - p > \Theta \tag{14}$$
$$\text{with} \quad l = -\alpha, \ldots, +\beta.$$

Here, $\aleph(i)$ is the onset detection function at point i. A thorough explanation of the method can be found in [5]. The parameters α, β, a, and b are chosen as standard values from related NOD work. The wait between peaks, Θ, is calculated from the STFT parameters and δ is tuned using a short piece of labelled data and the F1-score:

$$p = \frac{n_{tp}}{n_{tp} + n_{fp}}, \quad s = \frac{n_{tp}}{n_{tp} + n_{fn}}, \quad F_1 = \frac{2sp}{s+p}. \tag{15}$$

D. Results

To test the effectiveness of the proposed note onset detection method we test on a small dataset generated from the MAPS[1] dataset, from this we compose both polyphonic and monophonic pieces using the MixNotes[2] tool developed by Mina Mounir. Results range around an F-score of 0.9 for monophonic and 0.8 for the polyphonic pieces. Results on more realistic data on the whole AMT pipeline are presented in section V.

V. Automatic Music Transcription: Non-Negative Matrix Factorization

Using the results from section IV, we then do the second step of the AMT.

[1]*Midi Aligned Piano Sounds* dataset, freely available under Creative Commons license

[2]https://gitlab.esat.kuleuven.be/dsp-public/mix-notes-mina-mounir/-/tree/master

A. Non-Negative Matrix Factorization

Non-negative matrix factorization (NMF) was first proposed by Lee and Seung [9]. The idea is to represent a non-negative matrix X as a product of two non-negative matrices W and H, such that an error function is minimized.

$$C = ||X - W \cdot H||_F \tag{16}$$

Where

$$||.||_F$$

is the Frobenius norm. The matrices have the following dimensions.

$$X \in \mathbb{R}^{M \times N}, W \in \mathbb{R}^{M \times R}, H \in \mathbb{R}^{R \times N} \tag{17}$$

The parameter R sets the rank of approximation, thus called the reduced rank. It controls the power of summarization. If R=M we can see that matrices W and H will contain the exact decomposition of X and they will not be so informative to us. As the value of R is decreased, the matrices W and H start to take the values that best describe the main components in the matrix X. The intuition here is to see that NMF summarizes the rows of X in the rows of H, and the columns of X in the columns of W. In the application of AMT, the matrix X is generally a time-frequency information of an audio file.

B. NMF on magnitude spectrum

Generally, the magnitude spectrum of short-time Fourier transform is used to get the time-frequency information, namely the spectrogram. The horizontal axis of the spectrogram contains temporal information whereas the vertical axis contains spectral information. Therefore, in more common terms, columns of W describe the summarized notes in the recording whereas the corresponding rows of H describe the activations of a specific note that is summarized in the columns of W, given that the reduced rank R is well selected. Ideally R is set to number of notes in the recording. The selection of R is challenging, because it corresponds to how well NMF summarises the information in X. Though, there is not a clear guide on how to select the reduced rank R in AMT applications, it should be selected as the number of notes in the inspected recording. This is challenging to know since we do not possess the number of notes before we analyse the recording. In this paper [15] the authors estimated the reduced rank R as the number of onsets in the recording. They have achieved the number of onsets by applying an onset detection algorithm to the amplitude spectrum of the STFT spectrogram prior to the NMF algorithm. In our implementation we have used a similar approach to approximate as seen in the following Fig. 5.

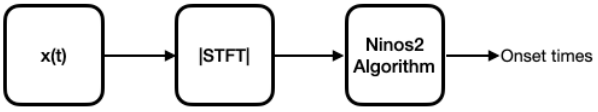

Fig. 5. Onset detection flowchart

For large recordings the audio file is divided into segments and fed through the system. The NINOS^2 algorithm described in section IV is used to detect the onsets and the number of detected onsets is set as the reduced rank R. Later, the estimated R is fed to the NMF, and factorization is performed on the magnitude spectrum of STFT. W and H matrices are estimated using the decompose function of Librosa [16]. For each column of W, which contains the spectral profiles, the maximum peak of the spectral profile is selected. Similarly, for each row of H, which contains the temporal information of the corresponding note, the largest activation is selected and matched with the onsets that were given by the NINOS^2 algorithm. Fig. 6 describes the system. After these steps, each

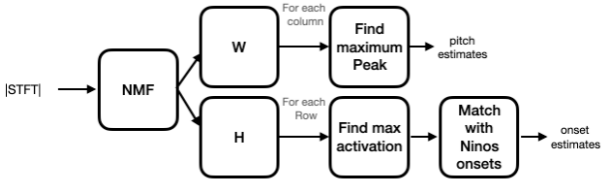

Fig. 6. NMF flowchart

note duration is estimated as the time between two consecutive onsets. Then, a list of note information is created and ordered based on the onsets of the respective notes. This list is a 2-dimensional list containing the following element structure.

[onset (s), note (Hz), duration (s)]

Finally, the music21 [17] library is used in python to convert the information to a music sheet.

C. Results on recordings

In this part we will demonstrate an example of a piano recording containing isolated notes, some results from a music therapy session, and finally we will discuss some issues that arise with the described AMT system in this section. We will first observe the estimated components and activations on isolated notes from a piano recording and then move on to the analysed section of the music therapy session. In the session the instructor kept the beat, and the students followed one by one by, tapping on the drums that were in front of them.

1) Isolated piano notes: In this piano recording, only 5 notes are played starting from C4 until G4. The estimated components and activations matrices from the algorithm are shown in Fig. 7 Upon examination we see that NINOS^2 algorithm estimated 5 onsets and this number is then used to estimate how many notes were in the recording. To create the magnitude spectrogram, we used a 2048-point STFT with a Hann window. The spectral profiles are informative, giving estimated frequency of the notes as 258, 290, 333, 344, 387 Hz respectively. When converted to the nearest piano note we have: C4, D4, E4, F4, G4 which corresponds to the recording. The Hz frequency estimates deviate from the exact frequency of the note due to quantization by the STFT.

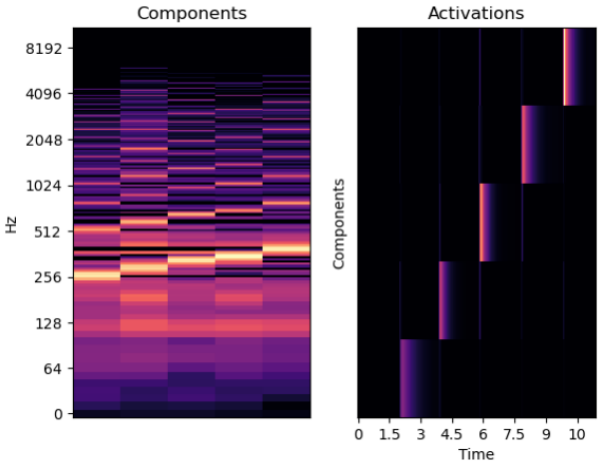

Fig. 7. Components and Activations of the recording

In this case the number of onsets exactly corresponded to the number of notes played in the recording, however, this might not be the case for every recording. In the case where we have smaller R than the number of notes, we do not have enough expressive power to describe the contents of the musical recording. If we have a larger R, then we try to estimate spectral profiles that are not present in the recording. Which makes post-processing harder in terms of deciding which component is a note and which one is not.

2) Results of the therapy session: The selected section from the music therapy session is 130 seconds long. This recording is divided into segments of 2 seconds and fed into the algorithm. The following describes the results we have obtained in comparison with the labelled data. The system predicted 215 onsets and the labelled data has 211 onsets. Out of these 215 predicted onsets, only 110 of them were in 0.1 precision of the corresponding labelled onsets. In these 110 onsets, 12.8% of the note estimates were accurate in comparison with labelled notes and 36.6% of the note estimates lied in $\pm 20Hz$ range of the labelled notes. As for the offsets our system was 7.9% accurate with respect to labelled data. The overall performance is summarised as follows.

Onset accuracy	51.16%
Note accuracy	11.26%
Offset accuracy	4.04%

D. Conclusion on the method

As can be seen from the results of the therapy session, the method did not perform good compared to isolated notes played on the piano. There are three reasons we were able to identify why the proposed system did not perform well. Firstly, our method cannot perform polyphonic transcription and the labelled data contained polyphonic sections where a student and the instructor played at the same time. The decision to select the highest activation in H matrix and matching it with the output of the NINOS^2 algorithm makes sure that we only estimate one note per onset. It can be easily the case where one onset may contain more than one notes, at which our

method fails to identify. Secondly, we assume that the note duration is the time between two consecutive onsets. This might not be the case, especially with drums, where hits follow one after another. This could explain why we have a bad accuracy on predicted offsets. Thirdly, we assume that the fundamental frequency of a note is at the peak of the spectral profile. In our application this assumption does not look like a bad assumption. We suspect that, either the labelled data contain errors, or due to sensitive nature of NMF to noise, it is not giving an accurate model for the notes in the recording. Knowing that frequency templates evolve with time and amplitude, reducing a note to a single frequency template in NMF could be the limiting factor in note accuracy of this AMT system.

VI. Combined system

A microphone array is used to pick up the sounds in the music therapy session. Namely, 8 microphones were arranged in a circular array. This means that each microphone outputs a stream of audio. Then these streams are used as an input to the BSS. ICA is used to separate the signals based on their statistical properties. After BSS, we have a stream of audio for each instrument (ideally). Then, these recordings are put through the NINOS^2 onset detection algorithm, described above, to get a reduced rank estimate (or number of onsets in this case) for the following method. NMF is then performed on each stream of data producing onsets, offsets, and note estimates.

VII. Conclusion & future work

This paper represents an approach which utilizes a combination of blind source separation, note onset and non-negative matrix factorization to identify and classify musical recordings. The pipeline is structured such that input recording goes into blind source separation and then into note onset detection and then into non-negative matrix factorization. No meaningful performance analysis can be made about our blind source separation method due to the obstacles such as the lack of labeled data, non-linearity in the mixing and implementation mistakes in our Python script. Note onset detection using NINOS method tested on a small dataset generated from the MAPS dataset which contains both polyphonic and monophonic music pieces resulted around an F-score of 0.9 for monophonic and 0.8 for polyphonic. The performance analysis of non-negative matrix factorization is made with two different recordings: a recording of a piano recording containing 5 isolated notes starting from C4 until G4 and a recording of 130 seconds long music therapy session. The result obtained from our algorithm is compared with labeled data. The number of onsets exactly corresponded to the number of notes played in the piano recording, however, this is not the case for music therapy session recording. For music therapy session recording, our algorithm has 51.16%, 11.26% and 4.04% for onset, note and offset accuracy respectively. To sum up, our approach performs better on recordings of isolated instruments than polyphonic music recordings.

Future work can focus on instrument recognition to estimate which instrument corresponds to which source in the recording. It is also interesting to approach automatic music transcription in a different way by making it genre specific instead of generalizing all genres, by this way more powerful models can be used in estimation of prediction of results. Another possible approach is by using machine learning/neural networks, however, machine learning approach is usually treated more or less as a black box model.

References

[1] S. Makino, S. Araki, R. Mukai, H. Sawada, and H. Saruwatari, "Ica-based blind source separation of sounds," *Proc. JCA2002*, pp. 83–86, 2002.

[2] P. Xu, Y. Jia, and Z. Wang, "Blind source separation using fast-ica with a novel nonlinear function," *arXiv preprint arXiv:1907.03432*, 2019.

[3] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," vol. 13, no. 5, pp. 1035–1047.

[4] S. Böck and G. Widmer, "Maximum filter vibrato suppression for onset detection," p. 7.

[5] M. Mounir, P. Karsmakers, and T. van Waterschoot, "Musical note onset detection based on a spectral sparsity measure," vol. 2021, no. 1, p. 30. [Online]. Available: https://asmp-eurasipjournals.springeropen.com/articles/10.1186/s13636-021-00214-7

[6] J. Schluter and S. Bock, "Improved musical onset detection with convolutional neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6979–6983. [Online]. Available: http://ieeexplore.ieee.org/document/6854953/

[7] A. Lacoste and D. Eck, "A supervised classification algorithm for note onset detection," vol. 2007, no. 1, p. 043745. [Online]. Available: https://asp-eurasipjournals.springeropen.com/articles/10.1155/2007/43745

[8] S. Böck, A. Arzt, and F. Krebs, "Online real-time onset detection with recurrent neural networks," p. 4.

[9] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[10] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No. 03TH8684)*. IEEE, 2003, pp. 177–180.

[11] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *IEEE Transactions on neural Networks*, vol. 17, no. 1, pp. 179–196, 2006.

[12] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," Nov. 1901. [Online]. Available: https://doi.org/10.1080/14786440109462720

[13] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in statistics*. Springer, 1992, pp. 162–190.

[14] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[15] S. H. Park, S. Lee, and K.-M. Sung, "Automatic music transcription using non-negative matrix factorization," in *Proceedings of 20th International Congress on Acoustics, ICA*, 2010.

[16] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015, pp. 18–25.

[17] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," 2010.