ASEAN Journal of Scientific and Technological Reports

Research article

AI System Design for Robotic Hand to Play the Piano

Wuttichon Aukkhosuwan[1*] and Wannarat Suntiamorntut[2]

[1] Faculty of Engineering, Prince of Songkla University, Songkhla, 90110, Thailand; 6210120041@psu.ac.th
[2] Faculty of Engineering, Prince of Songkla University, Songkhla, 90110, Thailand; suntiamorntut@gmail.com
* Correspondence: 6210120041@psu.ac.th;

Abstract: Robotic and Artificial Intelligent (AI) have been introduced as a key factor for industry revolution4.0. Many industries, such as manufacturing, agriculture, logistics, supply chain, and so on, are transformed and applied robotic and AI to enhance productivity and reduce cost. AI in creative work is very challenging, especially in music. This paper presents a system to enable the robotic arm to play piano notes with minimal errors. We used the knowledge of Optical music recognition (OMR), Automatic music transcription (AMT), Music source separation (MSS), and the elimination of robot arm cycle times problems for creating this system. The robotic arm used in testing with this system was LEGO. It can perform 4 functions. The first function is to play piano notes from sheet piano using Sheet Vision and Tesseract-OCR. Note reading accuracy is 21.36%, and note reading accuracy with note duration is 13.46%. The second function is to play the piano like a piano sound separate from a music file using Spleeter and Onsets and Frames. Piano note accuracy is 58.32%, piano note onset time error is ± 0.17, and piano note duration time error is ± 0.65. The third function is to play piano notes from real-time piano sounds using Onsets and Frames Real-time mode. The last function is to play the piano notes from the brain waves by comparing the frequency of the brain waves with the frequency of the piano notes. The design and experimental results are explained in this paper.

Keywords: Artificial Intelligent; Robotic; Music Source Separation; Automatic Music Transcription; Optical music recognition; Electroencephalographic

Citation:
Aukkhosuwan, W.; Suntiamorntut, W. AI system design for robotic hand to play the piano. *ASEAN J. Sci. Tech. Report.* **2022**, *25*(3), 59-68. https://doi.org/10.55164/ajstr.v25i3.246950.

Article history:
Received: July 1, 2022
Revised: September 27, 2022
Accepted: September 28, 2022
Available online: September 29, 2022

1. Introduction

Music has been a part of human culture for thousands of years. Apart from its entertainment value, music could improve people's mental and physical health. Artificial Intelligence (AI) was introduced in 1956 and had been developing from time to time. AI and creative works are very challenging, especially AI and music technology. AI in 2020 enables machine-to-human learning and problem-solving skills. Many music production tools are developed, allowing users to create beats and lyrics with a new musical experience easily.

The work of a musician career involves extracting the music notes sound from a song and trying to play them like the music notes they listened. Another task of the musician is to read music notation sheets. Musicians must play the right chords and notes at the right time.

Playing live music is very popular these days. However, people can listen to the songs of their favorite artists on many channels. But everyone still yearns for live music at restaurants or concerts. Sometimes, live music performances have issues about musicians' illnesses, accidents, or other reasons that prevent musicians from appearing on stage. Moreover, each time, the cost of hiring musicians to perform live music is high. If a restaurant has live music daily, the cost will be high. So, if we have a robotic assistant that can act as a substitute for musicians for live performances on stage, it would be great because the robot will not get sick and will also reduce the daily expenses. These problems can be solved if it's a robot with low production costs but with similar capabilities as a musician. Playing live music by robots also attracts a new audience of customers passionate about technology.

Currently, robots are being used as assistants to musicians and performing live with musicians, for example, Shimon [1]. It's a robot that composes songs according to the sounds it hears. Shimon used to perform live with the musicians. Shimon's live performance was something new and interesting to the audience. But Shimon was quite expensive to create. In addition to Shimon, there is much research about designing robotic arms to play musical instruments [2 – 5]. But designing and building these robotic arms is difficult for the average person and quite expensive. Therefore, the researchers wanted to create a system for the LEGO robotic arms to make the robot arms have the ability as close to the musician as possible. LEGO robotic arms are robotic arms that anyone can assemble, even without robotic knowledge. LEGO is not very expensive compared to other robotic arms.

The instrument the researchers chose for the system testing, and playing of the robotic arms was the piano. The piano is a simple instrument and can be performed live solo without needing another instrument.

Research on Optical Music Recognition (OMR) started a long time ago. It's the task of detecting musical characters on the five lines of the music note sheet. In 2014, a research paper used a camera to read music note sheets from paper instead of music note sheet files[6]. This research uses a method to detect bars in the music note sheet. Then zoom the camera to see notes in bars. This method is more efficient at reading music notes than reading notes from a file (depending on camera quality). But it's more costly than reading music note sheets from a file. Therefore, the researcher chose a method to read the music note sheet from files because it uses a lower cost. But the camera reading method is also an interesting alternative.

Many works in music are continually growing, such as eliminating noise and music generation. Music source separation (MSS) and automatic music transcription (AMT) are the techniques required for this study. The MSS carries out the separation of different sound sources in a song that can separate the sound of other instruments and the vocals from the music. There are many powerful MSS tools available today[7–9]. But there are a few tools that can separate piano sounds from music. Most instruments can only distinguish vocals, bass, drum, and other sounds. So Spleeter[9] is our choice.

Automatic music transcription is an audio signal analysis that produces a written transcript of a musical piece by identifying the instruments and notes played. There are many tools for automatic music transcription[10–12]. Test results show that Onsets and Frames [10] are highly efficient and easy to use. Moreover, it can work in real-time. It was therefore chosen as a tool that we used.

This work presents an AI system that can help a robotic arm to play the piano according to the music piano sound input. It can make robots the ability like musicians. Moreover, we've also added the ability to turn brainwaves into notes to provide a new alternative for live music. The tools that are used to create our system are as follows.

A. Spleeter

The Spleeter is a tool used to separate the source of the music. It contains pre-trained U-nets called the 5-stems model. The 5-stems model separates vocals, bass, drum, piano, and other sounds. The architecture of the U-nets is an encoder/decoder Convolutional Neural Network (CNN) with skip connection layers. It uses 12-layer U-nets (6 layers for the encoder and 6 for the decoder).

B. Onsets and Frames

The Onsets and Frames is a polyphonic piano music transcription tool. It uses deep convolution and recurrent neural networks to create models for predicting onsets and frames. The model was pre-trained before being used for prediction. The f1 score of the note with offset detection of Onsets and Frames is 50.22. Onsets and Frames can now process in real-time.

C. Optical Character Recognition

Optical Character Recognition (OCR) is an electronic process for translating text images into the computer's editable text. The OCR's performance depends on the image's quality for processing. This research uses OCR to convert music sheets (image files) into MIDI files.

D. Tesseract-OCR

Tesseract is an optical character recognition engine that detects text in images and converts it into computer-editable text. It is a tool developed by Google. In version 4, the Long Short-Term Memory (LSTM) mechanism and model have been added, enabling Unicode (UTF-8) support and recognizing up to 116 languages.

E. Music Instrument Digital Interface

Music Instrument Digital Interface (MIDI) is a protocol used for communication between electronic musical devices. The MIDI files store instruction sets for operating an electronic musical device. Parts of the instruction set contain controls to play the notes, the ordinal number of the musical notation, the volume, and the duration.

F. MindWave Mobile 2

MindWave Mobile 2 is a Bluetooth brainwave reader tool based on the TGAM1 module. The output of the sensor is 12-bit raw brainwaves (3-100Hz) with a sampling rate of 512Hz, EEG power spectrums (alpha, beta, etc.), NeuroSky eSense meters (attention and meditation), and eye blinks.

2. Materials and Methods

The robot arm already tested with this system is the Lego Mindstorms EV3. The main components used to build the robot arm to test the system are seven large motors, two EV3 bricks, as shown in Figure 1, one medium motor, and two Edimax EW-7811Un, as shown in Figure 1. The large motor is used to control the finger press of the robot arm, as shown in Figure 2. One large motor can now control two fingers at different times. EV3 bricks are clients that receive server commands to run the motor. One EV3 brick can control only four motors. The medium motor is used as a left and proper motion control on the belt of the robot arm, as shown in Figure 2. The Edimax EW-7811Un is a wireless adapter used to connect to the internet of the EV3 bricks.

Figure 1. Shows EV3 brick (left) and Edimax EW-7811Un (right).

Figure 2. It shows the robotic arm that contains seven large motors (left) and a medium motor (right).

Only four variables in the MIDI file were interested while LEGO was playing music. They are music notes, offsets (or duration), onsets, and velocities. Before playing the song, LEGO has been ordered to move to the octave, where the music notes are played. The note variable holds the information of the note played, with the onset as the start time and the offset as the stop time. The velocity in the MIDI file stated the speed of pressing a musical note. The fast pressing of the notes makes the sound of the notes louder. The value of this variable is sent as the motor speed for pressing the musical notes to increase the music's dynamic. Adding dynamic to the song makes the song more beautiful. The robot arm is still unable to press the sharp and flat note. The system needed to change the music key to C major before sending it to the robot to play.

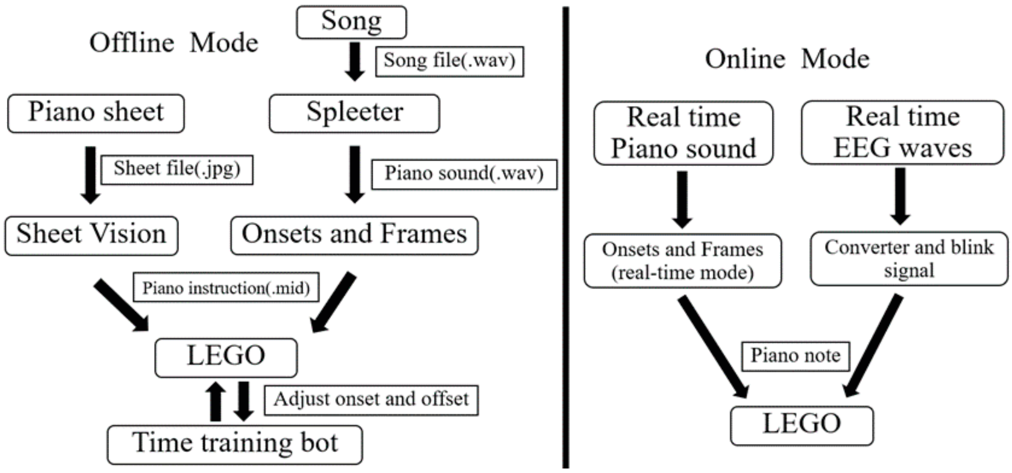

Figure 3. Shows the system processing diagram.

The built system can do 4 things. The first is that the system can read the brain waves of the wearer Mind Wave Mobile 2. After that, the system turns the read data into a series of notes and sends it to LEGO to play in real-time. It is a prototype and a method of sending brainwave data for those interested in further developing music playing through brainwaves.

Second, the system can turn the piano sound heard into piano note data. The system transmits piano notes to LEGO to play the piano in real time. It's useful for live music and piano copy shows. It can be improved by adding the song predictions function of the piano sound detected. In the end, it can be developed into a robot performing live with musicians and can play together at the same time.

Third, the system can separate the piano sound of the song. Then turn the piano sound into a piano note dataset and send it to LEGO. This will allow us to know the note and chord of the new song that does not have public information of note and chord. It is a skill that musicians must have. In the end, we don't need to add piano notes of the new song to the system. Just insert a new music file into the system. LEGO will be able to play piano.

Lastly, the system can convert the piano sheet files (JPEG files) to piano note data. Then send it to LEGO. It is a skill that every musician must have. That is to read the music sheet and play the notes in the correct rhythm. LEGO has a limitation regarding moving speed. The operating time of each motor is 0.7 s. It is average time it takes to move (test move 30 times).

Moreover, the moving speed of the motor will gradually decrease with increasing service life. So, these motors of LEGO cannot play songs with more tempo than 85 bpm. The tempo is the speed at which a musical note is played, usually using beats per minute (bpm).

From Figure 3, the system it has built can perform two functions. The first function is the online mode (real-time). The first step is to import real-time piano sound. Second, use Onsets and Frames to transcribe the received piano sound and convert it to piano notes. After receiving the musical note data, the system transformed it into a LEGO instruction set, as shown in figure 4. The last step is to send an instruction set to LEGO to play music notes.

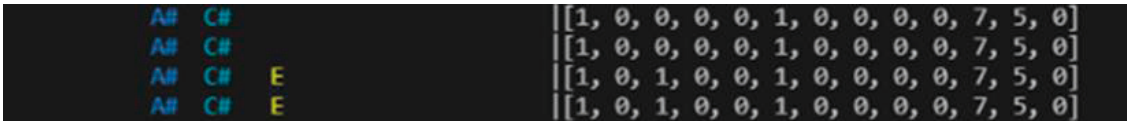

Figure 4. Shows the music notes detected from real-time piano sound by Onsets and Frames (left), and the message has been sent to LEGO (right).

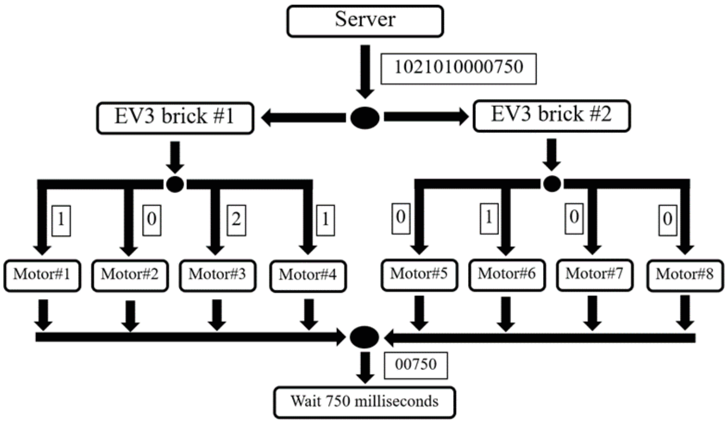

Figure 5. It shows a diagram of operating the LEGO motors from the server.

In Figure 5, the server has sent instructions to both EV3 bricks. The EV3 bricks have sent commands to each motor based on its position. The first seven motors are used to control the press. Code 0 is the order to do nothing—codes 1, 2, 3, and 4 spin forward to command the first finger to press. The higher the number, the higher the speed. Pressing an electric piano at high speed will make the sound even louder. Pressing it slowly makes the sound softer. It is to control the weight of each music note for a more melodious song. Codes 5, 6, 7, and 8 are to spin back to order the second finger to press. The 8th motor is used to move on the belt.

Code 0 is the order to do nothing. Code 1, 2, 3, and 4 are to order the robot arm to move to the left. The higher the number, the greater the distance. Codes 5, 6, 7, and 8 are to order the robot arm to move to the right. In the last step, the server ordered both EV3 bricks to wait. The last five digits are the waiting time. This function cannot use the time training bot. The operation of this function is in real-time. Therefore, it cannot reconfigure the onset and offset values before sending them to the robot arm.

The second function instructs LEGO to play music based on brainwaves read from the EEG sensor. The system reads brainwave data from the Neuro Sky Mind Wave Mobile 2 Headset. The system converts the read data into music notes in the specified frequency range. The system turns the music notes into LEGO instructions. The brainwave data used in the test are attention (any value read from the EEG sensor can be used). The note's frequency range is limited to two octaves. The command to instruct the robotic arm to press a music note is blink action. When the user blinks, the blink value read from the EEG sensor will be greater than 0 (if not blink, the value will be 0). When the blink value is greater than 0, the system sends information to LEGO for music note pressing, shown in figure 6.

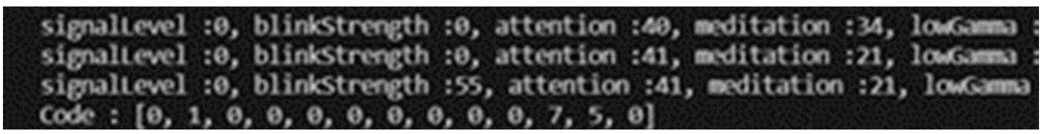

Figure 6. It shows the data from the EEG sensor and the code sent to the robot arm.

The second mode is offline mode. This mode can import two types of data, a JPEG piano note sheet, and a music WAV file. The piano sheet input solution uses a field Optical Character Recognition (OCR) tool called Sheet Vision. It is a python programming language developed by Calvin Gregory and Calvert Pratt. It detects musical characters such as notes, flats, and sharps in an imported piano sheet. Sheet Vision changes the result of detection to a dataset of musical notes. Sort music notes by time and export them as MIDI files.

This system has added tempo detection from piano note sheets using Tesseract version 4.1.1. Set the OCR engine to default and page segmentation mode to sparse text, and crop the image to make it easier to check. A way to detect tempo in a song is to extract the text with an equal sign in front of the number. The number after the removed equal sign is the song's tempo or speed.

For WAV file imports, the system sent imported data files to Spleeter. Spleeter separates various instrument sound sources using the 5-stems model and exports the separated instrument sound WAV files. The system sends WAV files of piano sounds to Onsets and Frames. It detects onsets and frames of the music note and exports it as a MIDI file.

MIDI files are converted to LEGO instruction sets and sent to LEGO. The robotic arm has the time it takes to move (cycle time). The cycle time distorts the time each note plays. Therefore, the last stage of the mode offline mode is a time correction of each music note. The time training bot runs this process. The bot records the robotic arm playing through the midi interface of an electric piano using a python library called Pygame. The robotic arms music playback is saved in MIDI file format. The new files resulting from playback are compared with the old MIDI files received. A different time is the cycle time of the robotic arm. The bot corrects the time in old MIDI files by reducing music note playback time with the cycle time of the robotic arm. With the bot's functionality, the robot can play the notes at the correct time of the song.

3. Results and Discussion

The data set for testing the performance of the WAV import system is The Lakh MIDI Dataset v0.1 (LMD) [13]. The LMD dataset is chosen because it has a large amount of quality data. It has piano tracks and is a popular MIDI data set. The total number of files tested was 5373, which is only part of the LMD data set. Every file is a MIDI file that contains a piano track. The data on the piano track analyzed were music notes, onset, duration, and velocity. The test aims to verify the performance of MIDI files generated by WAV files that Spleeter and Onsets and Frames have processed.

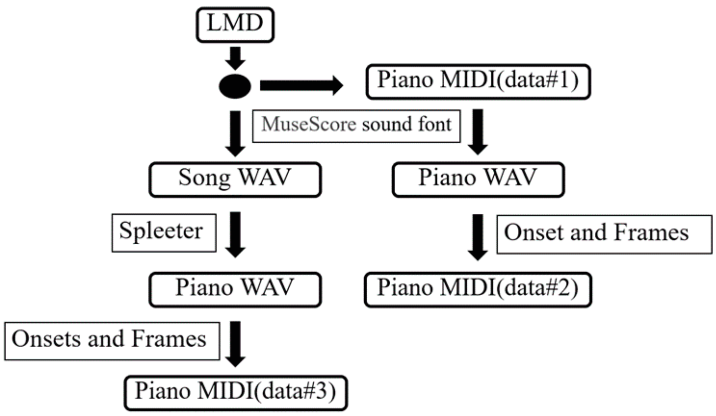

Figure 7. Shows data test set creation diagram.

From Figure 7, the first data set comes from creating a new file that copies only piano tracks from the original data set. The second data set is obtained by converting the MIDI files from the first to WAV files. The sound font converted is Muse Score versions 2.2 and 3 with GM (General MIDI). Use Onsets and Frames to convert WAV files back to MIDI files. This second data set was created to test the performance of Onsets and Frames. The third data set was obtained by a method like the second but used Spleeter to separate piano MIDI files. This third data set was created to test the effectiveness of Spleeter implementation. Finally, compare the first and second data set versus the similarity between the first and third data set.

The first step of the midi file comparison is to split the onset interval of the notes in the first MIDI data set. The upper boundary of the first note interval is in the middle of the first note and the second note onset time. It is the same value as the lower boundary of the second note interval. The lower boundary of the first note is double the onset time minus the upper boundary time. The upper boundary of the last note is double the onset time minus the lower boundary time.

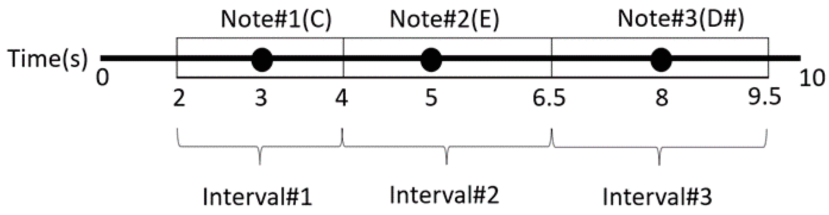

Figure 8. Show the segmentation of the onset interval of the musical notes in dataset 1.

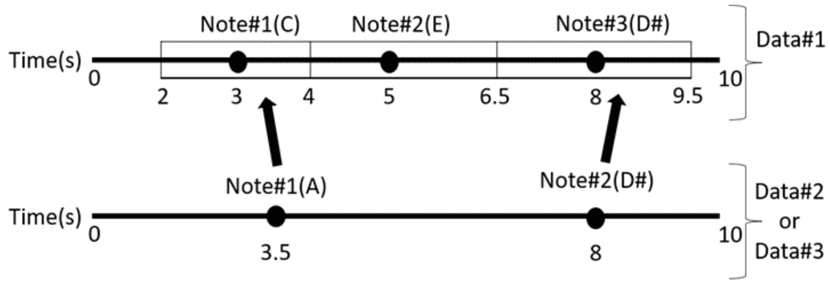

Figure 9. Show music note matching (paired by nearest time).

In Figures 8 and 9, The black point is the time when the music notes begin to sound (onset). The construction of the lower and upper bound of intervals depends on the start time of nearby music notes. If the lower bound of the first note is less than 0, it will be rounded to 0. The real reason for creating the onset interval is to eliminate the issue where some music notes are not detected. The onset interval help match music note for comparisons.

The music notes in the first and the matched second (or third) files are compared. MIDI files contain every music note's note type, onset, duration, and velocity information. Those data of the music notes were compared. They are weighted as one and tested for onset, duration, and velocity errors if they are the same note. But if they don't match, it's weighted as zero, and no tests are performed.

We used baseline statistics, weighted average, and error range to measure the experimental results' effectiveness. We examined the number of notes detected by the Onsets and Frames against the song's notes and displayed them as percentages. The accurate note of the measured music notes is compared to all the notes and is expressed as a percentage. As for the time discrepancy, the note starts playing (onset). We will calculate it as the interval of error. It then shows the range of potential errors. The duration of the note played and the volume of the notes are also checked similarly. We use this test method not to find the F1 score because we want details about the time the note starts, the duration of the press stop and the volume of the note press. This information can better control the robot in playing the piano notes.

Table 1. Show the results of Onsets and Frames and Spleeter tested

Onsets and Frames	Detecting (%)	Accuracy (%)	Onset errs (sec)	Duration errs (sec)	Velocity errs
Without Spleeter	91	75.72	± 0.025	± 0.51	± 24.93
With Spleeter	150.63	58.32	± 0.17	± 0.65	± 43.88

From Table 1, testing of Data Set 1 and Data Set 2, Onsets and Frames had a note detection average of 91 notes (input 100 notes). The accuracy rate was 75.72 percent. Accurate notes have an average onset time error of ± 0.025 seconds, a duration time error of ± 0.51 seconds, and a velocity error of ± 24.93. The test results of data set 1 and data set 3 (Spleeter was added), Onsets and Frames had a note detection average of 150.63 (input 100 notes). The number of extra notes caused by noise was averaged at 50.63. Of all the notes detected, the accuracy rate was only 58.32 percent. Accurate notes have an average onset time error of ± 0.17 seconds, a duration time error of ± 0.65 seconds, and a velocity error of ± 43.88. The MIDI velocity range is from 0–127.

This test also has issues with noise caused by Spleeter operation. It caused the note detection rate to be much higher than expected, and the accuracy was not as good as expected. The researchers speculated that what caused these problems was the size of the processing window of Spleeter. This is because selecting the Processing window affects the performance of music source separation [14].

The second tests the performance of reading piano sheet music and converting it to MIDI files by Sheet Vision. The dataset that was used to test was TheoryTab Database from Hooktheory. They are MIDI files and sheet music JPEG files. The total amount of data used in the test was 1160 files. This testing compares MIDI files output from Sheet Vision with MIDI files from the TheoryTab Database.

However, this process is to read the piano notes on the five lines from piano sheet music. The performance testing method compares the detected notes in pairs in sequence from two MIDI files and scores them shown in Figure 10. Scoring will be stopped when notes do not match, and subsequent notes will not be counted, as shown in Figure 11.

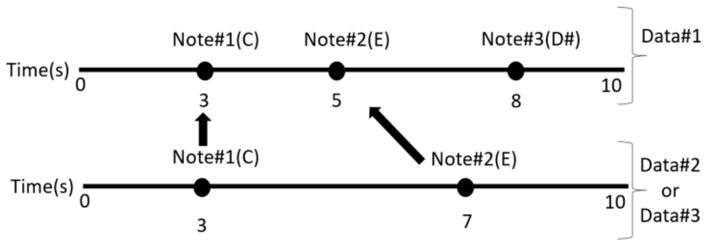

Figure 10. Show music note matching (paired by sequent).

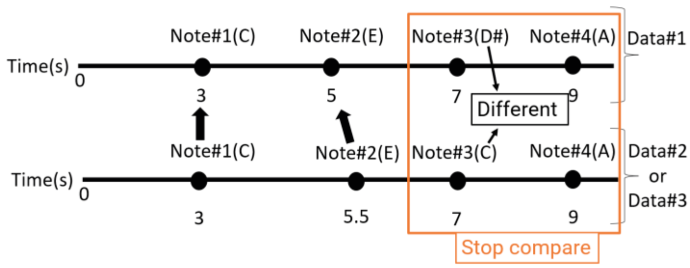

Figure 11. Show music note matching (paired by sequent) and stop scoring when the paired notes differ.

This test does not indicate how effectively detect notes and convert them to note values in a MIDI file. This test is based on a real robot arm's piano playing. If the robot misses a note, any subsequent notes are considered distorted.

Table 2. Show the result of testing

Tempo accuracy (%)	Note detecting (%)	Note accuracy (%)	Note with duration accuracy (%)
87.41	77.01	21.36	13.46

From Table 2, the result is the percentage of the correct interval size measured from the beginning of the sheet music. The note detection average of this system is 77.01 notes (input 100 notes). The accuracy of the note reading is 21.36 percent. The accuracy of the note reading with playtime duration is 13.46 percent. The tempo reading accuracy of this system by Tesseract is 87.41 percent.

The 2 functions only send data from Onsets and Frames (online mode) with hardware Mindwave mobile 2 to the system to convert to MIDI and then forward to LEGO. The performance data of both tools is pre-existing.

4. Conclusions

The system can read piano sheet music and WAV music files and export them as MIDI files. But the efficiency of reading piano notes is still low. The system can turn the resulting piano sound and brainwave data that reads from the EEG sensor into musical notes in real-time. The system can convert MIDI files and music notes into LEGO instructions. The system can also eliminate the excess time caused by the robotic arm's cycle times to play the electric piano.

Most of the limitations are in the part of the robotic arm that cannot play music at too high a speed. The robotic arm can't press the black electric piano keys (sharp and flat keys). The system needs to convert the music key signature into the C major keys before sending them to the robotic arm to play.

The system limitation for real-time conversion of piano sounds to piano notes is that it must operate in a quiet environment. Because the system may change the noise into piano notes, the system also can't extract only piano sounds of songs in real-time. So, the interesting future works are to build a system that can reduce noise and develop an AI that can extract piano sounds in real-time. Moreover, it is also interesting to apply this system to other instruments.

The future task of reading brainwaves if we can convert the brainwave signals of highly concentrated people into music. Then playing this song frequently to someone with ADHD may help them to improve their concentration. Moreover, music created by brain waves during activities may enhance the performance of the activity.

5. Acknowledgements

I would like to express my sincere thanks to my thesis advisor, Asst. Prof. Dr. Wannarat Suntiamorntut for her invaluable help and support throughout this research. Moreover, I would like to thank the Digital Research and Innovation Institute (DRii) for funding and research equipment.

In addition, I am grateful for Asst. Prof. Dr. Sakuna charoenpanyasak, Asst. Prof. Dr. Watcharin Kaewapichai and the researchers from Digital Research and Innovation Institute (DRii) for their suggestions and help.

Finally, I most gratefully acknowledge my parents and friends for all their support throughout this research.

References

[1] Weinberg, G.; Raman, A.; Mallikarjuna, T. Interactive jamming with Shimon: A social robotic musician. In: *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2009, 233-234. https://doi.org/10.1145/1514095.1514152.

[2] Lin, JC.; Huang, HH.; Li, YF.; Tai, JC.; Liu, LW. Electronic piano playing robot. In: *2010 International Symposium on Computer, Communication, Control and Automation (3CA)*. 2010, 2, 353-356. https://doi.org/10.1109/3CA.2010.5533457.

[3] Zhang, D.; Jianhe, L.; Beizhi, L.; Lau, D.; Cameron, C. Design and analysis of a piano playing robot. In: *2009 International Conference on Information and Automation*. 2009, 757-761. https://doi.org/10.1109/ICINFA.2009.5205022.

[4] Zhang, A.; Malhotra, M.; Matsuoka, Y. Musical piano performance by the ACT Hand. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, 3536-3541. https://doi.org/10.1109/ICRA.2011.5980342.

[5] Li, YF.; Chuang, LL. Controller design for music playing robot — Applied to the anthropomorphic piano robot. In: *2013 IEEE 10th International Conference on Power Electronics and Drive Systems (PEDS)*. 2013, 968-973. https://doi.org/10.1109/PEDS.2013.6527158.

[6] Fahn, CS.; Lu, KJ. Humanoid recognizing piano scores techniques. In: *2014 International Conference on Information Science, Electronics and Electrical Engineering*. 2014, 3, 1397-1402. https://doi.org/10.1109/InfoSEEE.2014.6946149.

[7] Luo, Y.; Mesgarani, N. Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation. *IEEE/ACM Trans Audio, Speech, Lang Process*. 2019, 27(8), 1256-1266. https://doi.org/10.1109/TASLP.2019.2915167.

[8] Defossez, A.; Usunier, N.; Bottou, L.; Bach, F. *Music Source Separation in the Waveform Domain*. Published online 2019.

[9] Hennequin, R.; Khlif, A.; Voituret, F.; Moussallam, M. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*. 2020, 5, 2154. https://doi.org/10.21105/joss.02154

[10] Hawthorne, C.; Elsen, E.; Song, J.; Roberts, A.; Raffel, C.; Engel, J.; Oore, S.; Eck, D. Onsets and Frames: Dual-Objective Piano Transcription. in *Proceedings of the 19th ISMIR Conference, Paris, France, September 23-27*, 2018. 50-57.

[11] Trabelsi, C.; Bilaniuk, O.; Zhang, Y.; et al. *Deep Complex Networks*. Published online 2017. https://doi.org/10.48550/ARXIV.1705.09792.

[12] Hawthorne, C.; Simon, I.; Swavely, R.; Manilow, E.; Engel, J. *Sequence-to-Sequence Piano Transcription with Transformers*. Published online 2021. https://doi.org/10.48550/ARXIV.2107.09142.

[13] Raffel, C. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. Columbia University, 2016.

[14] Sharma, S.; Mittal, VK. Window selection for accurate music source separation using REPET. In: *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*. 2016, 270-274. https://doi.org/10.1109/SPIN.2016.7566702.