REAL-TIME PIANO ACCOMPANIMENT MODEL TRAINED ON AND EVALUATED ACCORDING TO HUMAN ENSEMBLE CHARACTERISTICS

Kit ARMSTRONG[1], **Tzu-Ching HUNG**[1], **Ji-Xuan HUANG**[1], and **Yi-Wen LIU**[1]

[1]**Acoustics and Hearing Group, Dept. Electrical Engineering, National Tsing Hua University**, Hsinchu, Taiwan

ABSTRACT

With a view towards the goal of modelling the performance of classical music by human musicians, we have set our focus on the question of collaborative music-making in a MIDI environment. In previous work, we have presented a model that plays a part of a score in real time together with a live musician playing another part. We trained it to resemble human musicians faced with the same task, by tuning its systems built around a set of Kuramoto oscillators. Here we chose 3 musical works and conducted experiments collaborating with a variety of pianists and record the resulting performances as well as the testers' subjective impressions. We reconciled each performance with the corresponding music score, thereby defining a dataset which we call an "interpretation". In addition to subjective evaluation, we introduced objective criteria in the form of discriminants that classify interpretations as being the result of human-human interaction or of human-machine interaction. We considered the following qualities: desynchronization, jerkiness, and velocity curves. Our trained model performed similarly to humans with respect to the first two discriminants, but significantly differently with respect to the last. In light of this, it is notable that our experiment subjects often failed to correctly distinguish the two classes.

1. INTRODUCTION

Musical automata have exerted a fascination on musicians and music lovers for centuries. In addition to their use for recording and reproducing music, the possibilities of collaborative performance involving humans and machines together have received increased attention in recent years. Creating interactive capabilities in accompaniment systems has become a lively area of research [1].

An obvious first issue to be tackled concerns enabling a machine to recognize and process live human performance, in order to be able to interact musically with it. An important part of this is perceiving and "understanding" rhythm, for which various approaches to machine listening have proven fruitful, such as conducting post-factum probabilistic analyses of perceived onsets [2].

Our direction of focus is guided by the goal of creating an "AI musician", thus necessitating processing in real time. In the case of Western classical music, the nature of the task is shaped by the existence of a score, with which the performance is to be reconciled in a process called alignment or score-following. Effective methods have been devised for usage with audio performances [3] [4] or MIDI input [5]. Some primarily focus on the synchronization of timing, while others additionally capture various features of expression from human performances [6]. The development of synchronization systems has often been inspired by biological phenomena with little apparent connection to music [7]. In particular, the Kuramoto model [8] has been adapted and extended to describe observed aspects of collaborative music making [9].

The complementary part of real-time musical collaboration involves creating an output based on the information received from the human performer and on the indications of the score. Approaches range from modifying, or "time-warping" an existing audio recording to fit with the instantaneous timing of the particular performance [4], to generating a new digital performance of the accompaniment part from the score. Particularly germane to our present approach is the work of Cancino-Chacón et al. [6], which envisages a musician playing a physical piano that automatically accompanies him/her, responding to spontaneous changes in tempo, dynamics, and expression. In pursuing a similar goal, we have started from a more modular approach, by focusing first on the rhythmic aspect of collaborative music making.

In our previous work [10], we developed and trained a model on data that we collected from human musicians whom we asked to perform, in a controlled environment, a task identical to the intended application of the model. Our work thus does not take perfect synchronization as the ideal, but rather endeavors to imitate human responses. In this paper, we describe a set of experiments that were designed for testing the model's performance in playing along with human pianists. The rest of this paper is organized as follows: Sec. 2 defines our data structure for representing musical interpretation in the MIDI environment; Sec. 3 reviews the construction of the present model, and how it incorporates the Kuramoto coupling equations; Sec. 4 describes our experiments; Sec. 5 presents objective evaluations of their results; our experiment subjects' impressions are reported in Sec. 6; and discussion and conclusion are given in Sec. 7 and 8, respectively.

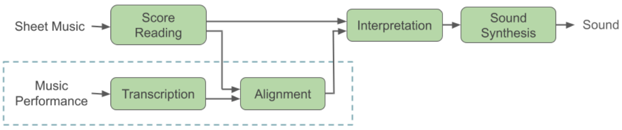

Figure 1. The role of the classical musician. The tasks within the dotted box apply when collaborating in an ensemble.

2. SCOPE

In this work, we represent scores as an ordered list of events, each following the format (type, note number, score position), where:

- "type" is "note-on" or "note-off";
- "note number" is the pitch designation according to MIDI;
- "score position" is the number of beats since the beginning of the piece.

The order of the list is determined by score position first, type second, and note number last. This method allows us to associate each relevant element of a score to a unique index number. In particular, each note is composed of a note-on event and a note-off event.

With our assumptions of a MIDI piano (excluding pedal) as our instrument, and error handling being outside of our scope, we can define an "interpretation" as the data of the score, as set out above, combined with information on how each event is performed. Specifically, to each event we associate the attributes of time and velocity. We may thus record an interpretation as an array of dimension $2N \times 6$, where N denotes the number of notes in the score, such that each row of this array follows the format of (index, type, note number, score position, time, velocity). This method also allows us to consider partial interpretations, where all score events are included, but not all events' attributes are defined.

In this framework, we may visualize the task of collaborative music making according to Figure 1.

3. MODEL

In this section we briefly recapitulate the essential aspects of our model. Its design is inspired by the application of Kuramoto oscillators [8] to biological phenomena [11], in particular to the synchronization of human tapping [12]. Although music obviously involves many complexities that do not appear in the referenced tapping task, we can make use of the same fundamental concept when we consider rhythm to be based on a more-or-less consistent beat, whose changes are subject to synchronization across players. We applied the model in the following environment: a human plays a one-voice part on a MIDI keyboard, while the model plays a second one-voice part, which we call here the "accompaniment" (though it may, in musical terms, instead represent the melody).

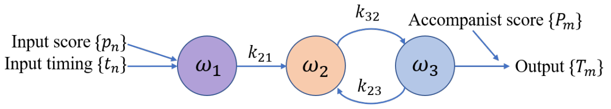

Figure 2. 3-oscillator Kuramoto model, as adapted for use in our model.

The model consists of 3 oscillators $\omega_1, \omega_2, \omega_3$ that are coupled as in Figure 2. Their positions are determined by the coupling equations:

$$\frac{d\theta_i(t)}{dt} = \sum_{j \neq i} k_{ij} \sin\big(\theta_j(t) - \theta_i(t)\big) + \Omega_i(t), \qquad (1)$$

for $i, j \in \{1, 2, 3\}$, where θ_i represent the positions of the respective oscillators, $\Omega_i(t)$ their intrinsic speed, and k_{ij} the coupling coefficients (with only k_{21}, k_{23}, k_{32} being non-zero, as per Figure 2).

For our application, since we treat music as consisting of discrete events, and no information is produced or exchanged between events, we must undertake certain non-trivial adaptations to the Kuramoto model. (See our previous publication [10] for details.)

By extrapolating the movement of ω_3, the Kuramoto oscillator system produces a series of predictions $\{T_m\}$ for the timings of subsequent events. Human instinct for the behavior that we are seeking to model served as the blueprint for integrating this system into our model. In particular, we presume that a human musician does not wait for each note before reacting; instead, he/she predicts how their partner will play, and how in consequence they should play, adjusting their prediction with every piece of new information that reaches their ears. We model this behavior by the following protocol:

1. the model makes a prediction of the timings of all future events up to a certain number of beats ahead;
2. the expected output based on this prediction is entered into a queue;
3. any time new information is received, a new prediction is made, and the following actions are executed:
   (a) output events that have already occurred will not be repeated, even if according to the new prediction they should be yet to occur;
   (b) output events that have not yet occurred, but that according to the new prediction should have already occurred, are carried out at once;
   (c) all other output events are updated in the queue according to the new prediction.

In this framework, it seemed reasonable to introduce a parameter t_r reflecting reaction time. That is, the above protocol is carried out after t_r has elapsed from the note having received. We denote by $\{T_m^*\}$ the timings of the output events hereby generated.

After performing a behavior-capturing experiment [10] with 20 subjects, for a total of 120 recordings, we searched for the parameters k_{21}, k_{32}, k_{23} and t_r that best fit the resulting data. Applying a gradient descent method over k_{ij}, t_r revealed a landscape with many local minima of similar value in the region of $5 \le k_{ij} \le 10$ and $t_r \approx 0.1$ seconds. We thus chose the values used in subsequent experiments to be $k_{21} = k_{32} = k_{23} = 5, t_r = 0.1$ seconds.

We programmed our model in Python, connecting it to MIDI input and output with RtMidi [1] . The program makes use of a clock (`current_time`), two queues (input and output), and 3 threads:

- `inputreading` records MIDI input events and their timing into the input queue;

- `calculating` matches each object as it appears in the input queue to the score, generating $\{t_n\}$, and computes θ_1 up until `current_time` by linear interpolation of the input onsets $\{t_n\}$. It then calculates θ_2, θ_3 according to Equation (1) and by extrapolating θ_3 produces a sequence $\{T_m\}$ of output timings;

- `worker` applies the protocol described above to output the notes at time $\{T^*_m\}$.

While error-handling is outside the scope of this research, we included a simple method to redeem minor mistakes. The `calculating` thread continuously keeps track of the last input note received, say, the n^{th} note. When it receives (from `inputreading`) the next event, it first checks if the pitch corresponds to $(n+1)^{\text{th}}$ or $(n+2)^{\text{th}}$ input note according to the score. In the former case, the action proceeds normally; in the latter case, the $(n+1)^{\text{th}}$ note is inserted with an identical timestamp. In any other case, the event is simply ignored. With this method, errors of the following nature — replacing a note with a wrong note, playing an extra note, or omitting a note — will only cause at most a local disturbance.

Furthermore, we implemented a simple method for following the human player's dynamics: with each new input, the program calculates the average of velocities of all (correctly played) input notes whose score position falls within the last beat at the time of calculation. This produces a running average v_n, to which we apply a linear transformation $V_n = \alpha v_n + \beta$ (the parameters determined by perceived musical preference – when the output represents a melody, we set α to be greater than when it is an accompaniment). All subsequent output notes are then provided with velocity V_n. Upon the arrival of the $(n+1)^{\text{th}}$ note, the output velocity will become V_{n+1}, and so on.

4. EXPERIMENTS

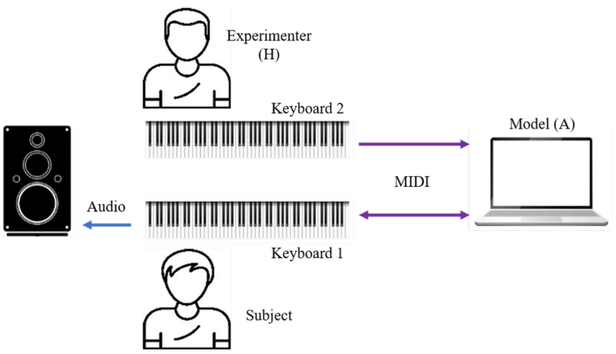

Figure 3. Hardware and human-interaction setup in the experiments

To test the performance of our model, we designed a test in which a collaboration between a human and our model could be compared in near-identical conditions to a collaboration between two humans. We set up two MIDI keyboards, and connected them to a computer which ran on macOS Catalina (version 10.5.7) with Intel Core i5 1.6GHz Dual-Core CPU and 4 GB RAM, as in Figure 3. All sound was routed through the computer's MIDI output, connected to a set of speakers. This setup allowed for the sounds to be indistinguishable, whether they are occasioned by an input on either of the keyboards or generated by the computer.

The experiment runs as follows: the subject is seated at keyboard 1 (KAWAI K-300 upright piano with AK-01 Touch Sensor), while an experimenter is seated at keyboard 2 (M-Audio ProKeys Sono 61), unseen by the subject. The subject is asked to play one part of a piece of music. The experimenter, without announcing their intention, chooses for the single trial in question either procedure A, in which the computer "plays" the second part by running the model, or procedure H, in which the experimenter plays it. [2] The subject, who has no knowledge of which case applies, is then asked whether he believes to have been playing with a computer or with a human. Listeners present in the room, but not looking at the operation of the experiment, are also asked whether they heard a human-computer or a human-human performance.

When the experimenter chose A, the program generated records of the following sequences:

$\{s_n\}$ the timing of the subject playing the input notes,
$\{t_n\}$ the input timing registered by the program,
$\{T^*_m\}$ the model-computed timing of output notes, and
$\{U_m\}$ the timing of the output actually being sent to the MIDI device,

and saved these in a series of `.txt` files. For H, another program was run, which simply recorded the inputs of both keyboards while transferring them for output through the speakers. For convenience of notation, we reuse the symbols $\{s_n\}, \{U_m\}$ for the timings of the subject and the experimenter, respectively.

For the above experiment, we selected 3 pieces of music:

1. W. A. Mozart: "Twinkle, Twinkle, Little Star" K.265, Theme

[1] https://spotlightkid.github.io/python-rtmidi/

[2] In order to avoid giving clues through external phenomena such as keyboard noise, in both cases the experimenter plays, but in the case A, his keyboard is not connected.

2. W. A. Mozart: "Twinkle, Twinkle, Little Star" K.265, Variation II

3. J. S. Bach/C. Gounod: "Ave Maria, Méditation sur le Prélude de Bach"

In (1) and (2), the participant was to play the right hand part while the computer (or experimenter) played the left hand. Each participant was asked to provide four extra notes at the beginning of each trial to indicate the intended tempo. In (3), the participant was to play the Prelude by Bach while the computer (or experimenter) played the melody by Gounod. To facilitate the task for the participants and minimize the confounding factor of their pianistic skill, for (1) and (2) they played a simplified version of the melody with only quarter notes.

We conducted the experiment with 12 participants, for a total of 80 trials. Each participant, with the exception of one, played each piece at least twice, and at least once with each partner (unknowingly).

5. DATA PROCESSING AND ANALYSIS

With the computer having recorded each trial, regardless of whether A or H was chosen, we proceeded to align the performances to the corresponding scores and produce "interpretations" in the form presented in Sec. 2. We constructed the following automatic protocol:

- First, we separate the two parts, in preparation for comparing them individually to the parts in the score.

- Starting from the first MIDI event, we attempt to identify the corresponding score event by searching for a match within a beat of the next expected score event. This means that if, for instance, at any moment the performer consecutively played wrong notes, or skipped all the notes, during a period of up to one beat, the next correct note would still be properly identified.

- Once the MIDI event is matched to the score event, the time and velocity of the former are written into the `interpretation` array, to produce an entry in the form (index, type, note number, score position, time, velocity) as required.

- If no match in the score is found, then the time and velocity are set to 0.

Each trial of our experiment thus yielded an input interpretation and an output interpretation. We rejected the trials with too many wrong notes, or where the above protocol failed for whatever reason to produce a correct alignment of score and interpretation. We proceeded to an analysis of these data, separately and jointly, in the form of discriminants designed to differentiate between A and H trials.

Piece	Number of trials	μ_Δ	σ_Δ
1	14	0.0656	0.0533
2	14	0.0420	0.0344
3	11	0.0254	0.0219

Table 1. Statistics of desynchronization in the A trials

Piece	Number of trials	μ_Δ	σ_Δ
1	13	0.0872	0.0385
2	13	0.0506	0.0921
3	11	0.0478	0.0684

Table 2. Statistics of desynchronization in the H trials

5.1 Desynchronization

The first question we investigate is whether between the machine and the human, one follows "better", i.e. more tightly. As stated in Sec. 1, our goal in making this model is not to create "perfect accompaniment", but rather "human-like accompaniment". Therefore this question, in the ideal case, should be answered in the negative.

To measure how well-synchronized the two parts are, we examine those notes which are meant to sound together according to the score. That is, we extract subsequences $\{\tilde{s}_k\}, \{\tilde{U}_k\}$ from $\{s_n\}, \{U_m\}$, respectively,including only the elements with indices n, m such that $p_n = P_m$. We define the desynchronization of a given trial as

$$\Delta = \frac{1}{l} \sum_{k=1}^{l} |\tilde{s}_k - \tilde{U}_k|, \tag{2}$$

where l is the number of the synchronization points according to the score (i.e. the number of elements in the sequences $\{\tilde{s}_k\}, \{\tilde{U}_k\}$). In practice, by force of extracting this information from the interpretation arrays, we only took into account synchronization points that were correctly played.

In the A trials, the statistics of Δ are shown in Table 1. The mean μ_Δ and standard deviation σ_Δ are calculated following the removal of trials where the computer suffered from unusually high lag (see Section 7). Similarly for the H trials, Table 2 shows the statistics following the removal of outliers containing too many wrong notes for identifying synchronization points to be practical.

As expected, the average desynchronization is highest for piece 1, as the score contains the least amount of information that a musician can make use of for synchronization. On the other hand, piece 3, with its running 16th notes in the accompaniment, allows the player of the melody to accurately predict the arrival time of the next beat in order to play their corresponding melody note with it.

It is interesting to note that the model, notwithstanding the fact that it was trained on human behavior, was in all cases slightly less desynchronized on average than the human.

Piece	Trials	μ_J	σ_J
1 A	19	0.719×10^3	1.405×10^3
2 A	22	2.817×10^5	2.556×10^5
3 A	24	0.716×10^2	1.355×10^2
1 H	17	3.074×10^2	4.069×10^2
2 H	11	1.205×10^5	0.947×10^5
3 H	17	0.699×10^2	1.546×10^2

Table 3. Statistics of total jerkiness in the A and the H trials

5.2 Jerk

It is sometimes said that machine performances sound jerky when compared to human performances. We tested this hypothesis using a traditional definition of jerk in mechanical movement as the third derivative of position (the second derivative being acceleration).

From the $2N \times 6$ interpretation arrays resulting from human and machine performances, we extracted the columns "time" and "score position" of note onsets, and computed the 3rd-order differences to generate a sequence $j_1, ..., j_{N-3}$ of jerk values. We then defined a measure of total jerkiness:

$$J = \sum_{i=1}^{N-3} j_i^2. \tag{3}$$

Table 3 shows the mean μ_J and the standard deviation σ_J of the total jerkiness in the A trials and the H trials for the three pieces, respectively. We indeed observe that for pieces 1 and 2, the computer accompaniment is jerkier than the human. The fact that this is not the case for piece 3 may be due to the nature of the voice part, which consists mainly of long notes. According to this measure, the model, with its stated goal of accompanying in a human-like manner, performed particularly well in piece 3. In Section 6, we see that A trials of piece 3 were often misidentified as H by participants and listeners alike.

5.3 Velocity Curves

As described at the end of Sec. 3, our model's output follows the velocity of its input according to a running-average-based method. We investigated whether humans, when performing the same task, exhibit a similar behavior. To this end, we supposed the MIDI velocity of each output note of an H trial, that is, each note played by the experimenter, to be a linear combination of the velocities of 11 preceding input notes. We then performed linear regression, with results shown in Figure 4.

We can see that the relationship between the velocities of the two voice parts is much less direct than assumed. For comparison purposes, Figure 4 includes the coefficients for piece 3 in the A setting, which are derived from the fact that since in this part each beat contains 4 notes, our model averages the 4 input velocities received during the previous beat.

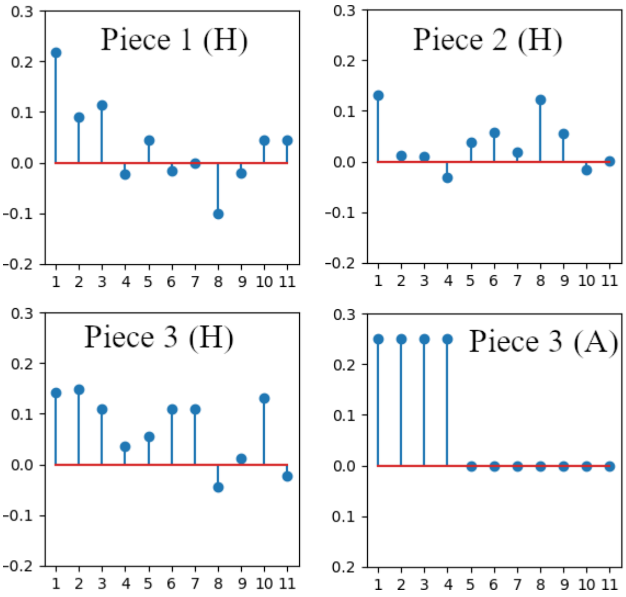

Figure 4. Regression coefficients for velocity prediction

6. SUBJECTIVE EVALUATION

Immediately following each trial, we asked the subject to guess the identity of the partner, and inquired about their reasoning and degree of confidence C (from 0 to 100) in the guess. Separated according to the piece, the results are shown in Table 4.

Piece	Trials	Correct	Success Rate	μ_C
1	28	18	64%	69.5
2	28	22	79%	67.1
3	27	9	33%	59.3

Table 4. Subjects' success rate and their mean confidence in guessing the identity of the partner (A or H)

We asked the same questions to listeners present, with the results shown in Table 5. These tables are tallied by trials and persons jointly; if two listeners were present at a trial, their opinions are counted as if one listener were present at two trials.

Piece	Trials	Correct	Success Rate	μ_C
1	14	12	86%	79.3
2	14	5	39%	75.0
3	10	1	10%	74.0

Table 5. Analogous data to Table 4 for the listeners

In total, 49/83 (59%) of the guesses by the performing participants themselves were correct, vs. 18/38 (47%) by the listeners. However, the listeners had higher average confidence in their guesses than the performing participants.

The fact that both groups' guesses for piece 3 were significantly worse than chance suggests the presence of false expectations. We experienced subjectively our model's performances of the Gounod melody as not sounding "mechanical" in the way that one might be preconditioned to

Piece	μ	max	σ
1	0.0101	0.0620	0.0068
2	0.0105	0.0942	0.0069
3	0.0116	0.2095	0.0088

Table 6. Statistics of the input lag (in seconds)

Piece	μ	max	σ_Δ
1	0.0052	0.0508	0.0070
2	0.0053	0.1131	0.0076
3	0.0041	0.0588	0.0061

Table 7. Statistics of the output lag (in seconds)

assume.

Though our focus is on the "musicality" of collaborative performance, other factors inevitably come into play when dealing with responses to such a general question as identifying a partner as human or machine. For instance, the synchronization of key noises to the accompaniment heard, despite our best effort to play along on a disconnected keyboard in A trials, might have served as a clue. The presence or absence of missing or wrong notes in the accompaniment also revealed itself as a telling difference, as errors were unique to the human partners. We tried to minimize this factor by playing our part carefully, and, in the event of being asked by the participant, suggesting that we possibly programmed the model to introduce random errors. Nevertheless, the participants might have drawn upon this rationale all the same to distinguish between our model and the human.

7. DISCUSSION

In the course of setting up our experiment, we determined by trial and error that the highest framerate at which our equipment was able to run the program smoothly was about $f = 100$/sec. That is, the output timing predictions (see Sec. 3) were updated 100 times per second. Note that the input and output timestamps were not quantized, but rather calculated with the maximum precision offered by our MIDI interface.

Nevertheless, even with the program running smoothly, non-negligible sources of lag affected both the input and the output; the program received the input events later than they occurred, and the output sent to the speakers was delayed with respect to the actual calculated output of the program. In the notation of Sec. 4, $t_n > s_n$ and $U_m > T^*_m$.

We measured the input lag $(t_n - s_n)$ and its mean, maximum value, and standard deviation across all trials of a given piece are shown in Table 6. Similarly, the statistics for the output lag $(U_m - T^*_m)$ are shown in Table 7.

As we see, the severity of the lag depended on the rates of input and output notes, the second piece being heavy in output and the third being heavy in input. Due to the construction of the model, input lag mainly affects the refreshing of the predictions (as prompted by the protocol described in Section 3), which while undesirable, is often unnoticeable since with a reasonably consistent tempo the new predictions are usually similar to the old ones. However, output lag has a more noticeable effect, as it directly delays what the listener hears. In the second piece, which requires up to approximately 10 output notes per second, notes were on occasion sounded up to 0.1131 seconds later than intended.

Our experiment subjects frequently commented on the feeling of being slowed down by their partner. It is reasonable to surmise that the lag described here is a significant contributor to this phenomenon.

A priori expectations regarding automatic accompanists play a significant role in experiments such as ours. Not only do they impact subjective evaluations after the experiment, they might also affect how the subjects play during the experiment. We have attempted to separate such effects through our "blind trial" design, in which the partner was chosen at random between a human and a machine.

We have chosen the repertoire pieces primarily based on their prevalence. That is, most people who have learnt the piano are familiar with the works in question, and potential problems of being unable to read or play them fluently were avoided. There exist surely classical works which, in a comparable test situation, would make the differences between human and machine performance more or less apparent. At one end of the spectrum, a piece where the part includes extended and important solos would lead to the performer being recognizable, and in particular, a machine being identified as such. On the other hand, a recitative-like accompaniment might be relatively easy to imitate. Indeed, this idea is upheld by the fact that our participants correctly identified the partner in the piece with the busy accompaniment (Mozart, Variation 2) much more often than in the piece where the roles are reversed (Bach/Gounod).

It is somewhat surprising that although the velocity-based discriminant presented in Section 5.3 identified an obvious difference between human and machine performances, it apparently did not have a bearing on the subjective evaluation by our participants. Further experiments may elucidate whether this aspect might be considered unimportant, or the present results were confused by unclear or diverse expectations. For example, if half of the participants consider that following their dynamic curve is a characteristic machine trait while the other half considers it to be characteristically human, analyzing the data as we have done would yield no result.

Compared with a real musician, our model only makes use of a small portion of the information theoretically available during a collaborative performance. Beyond the knowledge of the notes of a score, it has no intrinsic understanding of its musical content, simply continuing at a constant tempo and dynamic in the absence of continued external input. In fact, it only listens to the timing of its partner and, somewhat crudely, to their dynamics. Other aspects, such as the length of notes (often referred to as "articulation") as well as more complex paradigms of phrasing would appear intuitively desirable for natural-seeming musical collaboration.

8. CONCLUSION

In the context of creating an AI performer in a classical music environment, our model represents an attempt to capture certain aspects of how human musicians interact with each other during an ensemble performance. To this end, we have designed it starting from concepts reflecting our experience as musicians, and adjusted it using data gleaned from accompaniment and collaboration trials. When we set the trained model to play together with external experiment participants, we were able to collect information that allows us to analyze its performance, compared with that of humans recorded under nearly identical circumstances. As part of the experiment, participants and listeners were asked to distinguish, in a blind-trial-like setting, between human and machine. In this paper we presented some measures that quantify salient aspects of the quality of collaboration. A certain number of them, when applied to our data to evaluate the extent to which our model resembles human musicians, produce results that appear to be in accordance with, and perhaps explicative of the participants' responses.

Acknowledgments

This research was supported by the National Science and Technology Council of Taiwan under grant No. 111-2634-F-002-023.

9. REFERENCES

[1] C. Raphael, "A probabilistic expert system for automatic musical accompaniment," *J. Computational and Graphical Statistics*, vol. 10, no. 3, pp. 487–512, 2001.

[2] M. Leman, "Co-regulated timing in music ensembles: A bayesian listener perspective," *J. New Music Res.*, vol. 50, no. 2, pp. 121–132, 2021.

[3] S. Sako, R. Yamamoto, and T. Kitamura, "Ryry: A real-time score-following automatic accompaniment playback system capable of real performances with errors, repeats and jumps," in *Proc. 10th Int. Conf. Active Media Technology*, Warsaw, Poland, 2014.

[4] C. Raphael, "Music plus one and machine learning," in *Proc. 27th Int. Conf. Machine Learning*, 2010, pp. 21–28.

[5] P. Toiviainen, "An interactive MIDI accompanist," *Computer Music J.*, vol. 22, no. 4, pp. 63–75, 1998.

[6] C. Cancino-Chacón, S. Peter, P. Hu, E. Karystinaios, F. Henkel, F. Foscarin, N. Varga, and G. Widmer, "The ACCompanion: Combining reactivity, robustness, and musical expressivity in an automatic piano accompanist," https://arxiv.org/abs/2304.12939, 2023.

[7] E. W. Large, "Resonating to musical rhythm: theory and experiment," in *The Psychology of Time*. Emerald Group Publishing, 2008, pp. 189–231.

[8] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*. New York: Springer-Verlag, 1984.

[9] S. Shahal, A. Wurzberg, I. Sibony, H. Duadi, E. Shniderman, D. Weymouth, N. Davidson, and M. Fridman, "Synchronization of complex human networks," *Nature Comm.*, vol. 11, no. 1, p. 3854, 2020.

[10] K. Armstrong, J.-X. Huang, T.-C. Hung, J.-H. Huang, and Y.-W. Liu, "Real-time piano accompaniment using Kuramoto model for human-like synchronization," in *Proc. 16th Int. Symp. Computer Music Multidisciplinary Research*, Tokyo, Japan, 2023, pp. 744–747.

[11] J. A. Acebron, L. L. Bonilla, C. J. Vicente, F. Ritort, and R. Spigler, "The Kuramoto model: A simple paradigm for synchronization phenomena," *Reviews of Modern Physics*, vol. 77, no. 1, pp. 137–185, 2005.

[12] O. Heggli, J. Cabral, I. Konvalinka, P. Vuust, and M. Kringelbach, "A Kuramoto model of self-other integration across interpersonal synchronization strategies," *PLoS Comput. Biol.*, vol. 15, no. 10, p. e1007422, 2019.