The ACCompanion: Combining Reactivity, Robustness, and Musical Expressivity in an Automatic Piano Accompanist

Carlos Cancino-Chacón[1], **Silvan Peter**[1], **Patricia Hu**[1], **Emmanouil Karystinaios**[1], **Florian Henkel**[2], **Francesco Foscarin**[1], **Nimrod Varga**[1] and **Gerhard Widmer**[1]

[1]Institute of Computational Perception, Johannes Kepler University Linz, Austria

[2]SiriusXM + Pandora, USA

{carlos_eduardo.cancino_chacon, silvan.peter, patricia.hu}@jku.at

Abstract

This paper introduces the ACCompanion, an expressive accompaniment system. Similarly to a musician who accompanies a soloist playing a given musical piece, our system can produce a human-like rendition of the accompaniment part that follows the soloist's choices in terms of tempo, dynamics, and articulation. The ACCompanion works in the symbolic domain, i.e., it needs a musical instrument capable of producing and playing MIDI data, with explicitly encoded onset, offset, and pitch for each played note. We describe the components that go into such a system, from real-time score following and prediction to expressive performance generation and online adaptation to the expressive choices of the human player. Based on our experience with repeated live demonstrations in front of various audiences, we offer an analysis of the challenges of combining these components into a system that is highly reactive and precise, while still a reliable musical partner, robust to possible performance errors and responsive to expressive variations.

1 Introduction

Participating in ensemble music playing requires creative collaboration and can provide fulfilling musical experiences. Given their success in other creative domains, there is an emerging trend towards the exploration of AI systems in interactive and collaborative settings. Interactive accompaniment systems are computer programs that can perform jointly with human musicians. In such human-machine musical collaboration settings, the computer system can either improvise its musical content or render an expressive accompaniment according to a given score in response to the human player in real-time [Dannenberg, 1987; Dannenberg, 1984; Vercoe, 1984]. We target the latter case and we restrict our attention to the symbolic domain, i.e., we work with music data that, as opposed to audio files, explicitly encode high-level note features such as pitch, onset (when a note starts), and duration. Specifically, we propose the ACCompanion, an automated accompaniment system that, given a symbolically encoded score with a soloist and an accompaniment part, is able to render an expressive MIDI performance of the accompaniment in response to the MIDI performance of a human soloist. This system can work in different configurations, as long as the soloist performances can be encoded in MIDI data, and a MIDI player that translates MIDI outputs to sound is available (see a photo of the system in action on a computer-controlled player piano in Figure 1).

Figure 1: A photo of the ACCompanion in action using a computer-controlled piano. Some notes are played by the human soloist, and some by our system.

Since the development of the first MIDI accompaniment systems [Dannenberg, 1984; Vercoe, 1984], most research in this field has focused on the synchronization aspect, that is, matching the accompaniment with the human performance [Nakamura *et al.*, 2015; Chen *et al.*, 2014; Raphael and Gu, 2009]. In terms of musical expression in the accompaniment performance, most systems to date constrain their expressive response to individual expressive aspects such as tempo [Cont, 2008; Raphael, 2010], or dynamics and timing [Xia *et al.*, 2015]. We address the issue of musical expression in the accompaniment performance by explicitly modeling tempo, dynamics and articulation aspects into the ACCompanion, making it capable of generating expressive accompaniments in response to the performance style of the soloist. We also compare cognitive plausible tempo models inspired by the research in the sensorimotor synchronization.

Moreover, we have showcased our system in a number of real-world scenarios, including demonstrations to the scientific and general public (e.g., at the Falling Walls Science

Summit 2021 in Berlin, at Gerhard Widmer's Keynote at IJCAI-22 in Vienna and at the Heidelberg Laureate Forum Foundation). The ACCompanion won an Award for Creative Achievement at the AccompaniX competition organized by the Neukom Institute for Computational Science at Dartmouth College in 2017. However, our own experience also made clear that although our system can perform interactively and expressively with a human soloist, ensemble music performance and shared musical interpretations between humans involve a level of cognitive-emotional alignment that cannot yet be achieved. Given our experience, we want to start a critical discussion on the causes and possible solutions to this problem. Finally the ACCompanion will also be published as open source software.[1]

Overall, our contributions in the context of automatic accompaniment systems are as follows: (1) the generation of an accompaniment part that is conditioned on the soloist performance on three expressive parameters, that is, tempo, dynamic, and articulation; (2) the exploration of cognitively plausible tempo models, inspired by research from the field of sensorimotor synchronization; (3) an extensive system evaluation in various real-world scenarios, including concerts in public venues and critical exposure to professionally trained musicians. The rest of this paper is organized as follows: Section 2 presents related work and Section 3 outlines the ACCompanion system. The key components are evaluated in Section 4, followed by a human performer's perspective and a discussion on collaborative human-machine musical modeling in Section 5. Conclusions and future research recommendations are presented in Section 6.

2 Related Work

Dannenberg [1984] identifies three tasks that accompaniment systems must perform to play effectively with a human:

1. *Soloist part detection*: capturing a human performance in real-time and identifying the performed notes from its representation.
2. *Score following*: precisely matching these performed notes to notes in the score, while being robust against possible player errors.
3. *Expressive accompaniment generation*: producing an expressive accompaniment part that is conditioned on the soloist part.

The detection step is a prominent part of systems that work from audio recordings. Since we use MIDI data, this step is handled by the MIDI instrument, which explicitly encodes the note information. In this section, we present related work on the score following and expressive accompaniment tasks, as well as some relevant perspectives from the music cognition literature on joint music performance.

2.1 Score Following

The score following task can be performed in an online or offline setting, starting from MIDI or audio. In an online, MIDI-based context, we aim to perform *online symbolic score following*, aligning each note in a MIDI performance with corresponding elements (time or notes) in the musical score. Recent work by Raphael and Gu [2009] and Nakamura et al. [2014], both employ Dynamic Bayesian Networks for this task. Our ACCompanion employs a hidden Markov model (HMM) based score follower that is roughly comparable to the one by Raphael and Guo in design (see Section 3.1).

Related work has been developed in the domain of online audio-to-score alignment that aligns every temporal position in the performance with a temporal position in the score [Dixon, 2005; Cont, 2008; Raphael, 2010; Duan and Pardo, 2011; Arzt and Widmer, 2015]. These systems are based on On-Line Time Warping (OLTW) or variants of HMMs. A similar range of approaches, albeit in a non-causal formulation, is used for offline symbolic score following [Gingras and McAdams, 2011; Chen *et al.*, 2014; Nakamura *et al.*, 2017]. Recently, Peter et al. [in press] performed a systematic evaluation of dynamic time warping (DTW) versus HMM-based approaches for offline symbolic alignment, and found that they yield similar results for the alignment of piano classical pieces. DTW-based approaches have therefore the potential to be a viable (but still untested) alternative to HMMs for online symbolic score following, and they could have a different set of strengths and weaknesses. For this reason, in addition to the HMM mentioned above, the ACCompanion also contains an OLTW score follower, which adapts to the symbolic domain many ideas from the state-of-the-art in audio score following [Arzt and Widmer, 2015]. Note that the score follower is only a part of our system, and it is out of the scope of this paper to perform a systematic evaluation of the state-of-the-art in this field.

Gingras and McAdams [2011] use a tempo model to "smooth" erratic score follower outputs due to embellishments or player mistakes. Similarly, Raphael and Gu [2009] use a Kalman filter for this purpose and for predicting future note positions. We explore multiple variations of such a tempo model. Other recent systems [Arzt and Lattner, 2018; Agrawal and Dixon, 2019] suggest employing deep learning for enhanced audio feature pre-processing prior to alignment.

2.2 Expressive Accompaniment Generation

For generating an expressive accompaniment, a system needs to solve two problems. The first is to generate an expressive rendition of a musical piece that conforms to human understanding and perception of music and communicates its inherent emotional and affective content. The second consists of adapting this expressive rendition in response to the performance style of the soloist. Some work in the literature focus on the first problem alone with rule-based algorithms [Friberg *et al.*, 2006], probabilistic approaches [Widmer *et al.*, 2009], or, recently, neural networks [Jeong *et al.*, 2019; Maezawa *et al.*, 2018; Cancino-Chacón *et al.*, 2018]. A detailed description of these models falls outside the scope of the current paper; the interested reader is directed to [Cancino-Chacón *et al.*, 2018]. Note that all these systems work offline and are not suitable for a real-time scenario. However, they can be used for a first offline generation step, which gets then adapted to solve the second problem, i.e., to condition the generated ac-

[1] Supplementary materials and code for this paper can be found at the following link: https://cpjku.github.io/accompanion_ijcai2023/

companiment on the soloist performance.

Multiple works target this second problem but focus only on certain expressive performance parameters. Cont [2008] explicitly models the expressive tempo to enable the temporal interaction between the human performer and the accompaniment system, while not considering other expressive aspects, such as dynamics or articulation. Raphael [2010] uses a Gaussian graphical model as a tempo model to schedule the accompaniment performance, but otherwise constructs the performance itself by time-stretching a prerecorded audio recording of the accompaniment. More recently, Xia et al. [2015; 2015] proposed to model a duet ensemble as a linear dynamic system with learned parameters. At each position in the score, the time and dynamics of the next accompaniment notes are predicted as linear combinations of some local note features, "smoothed" by a hidden variable that acts similarly to the tempo model of Raphael.

2.3 Perspectives from Music Cognition

From a cognitive standpoint, we can consider an ensemble performance as a social event in which individual musicians are engaged in a joint action that results in feelings of musical togetherness. Such a joint action can be described in terms of the underlying sensorimotor synchronization (SMS), that is, the processes involved in the temporal coordination of an action or movement with an external rhythm [Repp, 2006].

Mathematically, SMS can be modelled via either the information-processing approach or via dynamical systems theory [Loehr *et al.*, 2011]. We follow the former, which assumes the existence of a timekeeper instance to count successive actions and generate motor commands in response to external stimuli. In the context of ensemble music performances, the rhythmic sensory stimuli are usually characterized by irregularities (expressive timing), resulting in some degree of uncertainty with regard to the precise timing of the next event. For the timekeeper to achieve sustained SMS despite this variability, some form of error correction and/or temporal prediction is necessary [Vorberg and Wing, 1996; Mates, 1994; Van Der Steen and Keller, 2013]. We consider these findings in our modeling choices for the underlying tempo models, which we describe in Section 3.2 and evaluate in Section 4.2.

3 System Architecture

Let us consider a musical score with a solo and an accompaniment part, and a human player performing the solo part. The musical score is assumed to be in a digital symbolic format (e.g., MusicXML, MEI) and the soloist is performing on a MIDI-capable instrument. We use Partitura [Cancino-Chacón *et al.*, 2022] and the Mido[2] for handling scores and real-time MIDI input, respectively. The ACCompanion generates a MIDI real-time expressive rendition of the accompaniment part that is conditioned on the tempo, dynamics and articulation of the soloist.

Figure 2 presents a schematic view of the system. Its main functionalities are implemented in two modules: the `Score Follower` and the `Accompanist`, which solve tasks 2 and 3 proposed by Dannenberg [1984]. These modules are the focus of the rest of this section. For practical implementation reasons, there is a third module, the `MIDI Handler`, which routes input and output MIDI messages. This module allows the ACCompanion to work in a variety of different configurations, including working directly with MIDI-capable instruments (e.g., MIDI controllers or player pianos) and being used with MIDI player software.

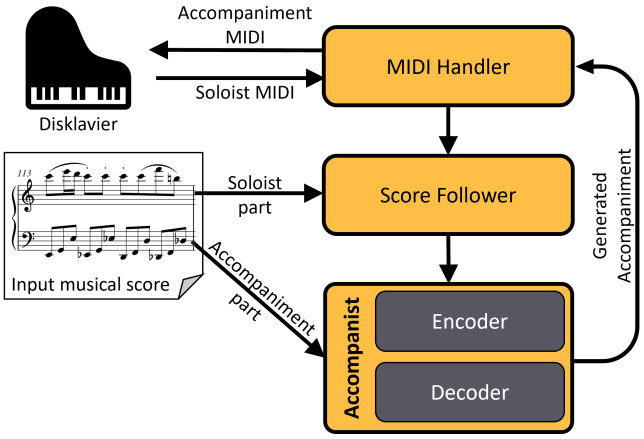

Figure 2: Modular architecture of the ACCompanion.

Before moving to a description of the `Score Follower` and `Accompanist` modules, we need to introduce some notation that is used in the rest of the paper. Figure 3 shows how score and performance information is represented in the system. We model the i-th score note with a triple $(p_i^{\mathrm{s}}, o_i^{\mathrm{s}}, d_i^{\mathrm{s}})$ which corresponds to its MIDI pitch, score onset time and duration in musical beats. Notes in a chord (i.e., notes are intended to be performed at the same time) will have the same score onset time, which we represent by $o^{\mathtt{score}}$. The time between two consecutive score onset times, the so-called inter-onset interval (IOI) is denoted by $\delta_i^{\mathtt{score}}$. The i-th performed note can be described by the quadruple $(p_i^{\mathrm{p}}, o_i^{\mathrm{p}}, d_i^{\mathrm{p}}, v_i^{\mathrm{p}})$, i.e., MIDI pitch, onset time and duration in seconds and MIDI velocity. We use $o_i^{\mathtt{perf}}$ to denote the performed onset time corresponding to the notes at score onset time $o_i^{\mathtt{score}}$.[3] The performed IOI is denoted by $\delta_i^{\mathtt{perf}}$.

3.1 Score Follower

Following our discussion in Section 2.1, we implement two alternative score following approaches, one based on HMMs and the other using OLTW. The score follower also contains a *note tracker* object that keeps a record of the MIDI velocity v^{p} and duration d^{p} of the past performed notes.

In order to deal with polyphonic MIDI inputs, we use a windowing approach: incoming input MIDI messages are aggregated into non-overlapping $10\mathrm{ms}$ windows, and we consider that all of the played notes inside each of these windows correspond to the same position in the score (i.e., belong to the same chord) and have the same performed onset

[2]https://github.com/mido/mido/

[3]Note that this onset time could be different than the onsets of the individual performed notes (as shown in Figure 3), since the notes in a chord are never played at exactly the same time.

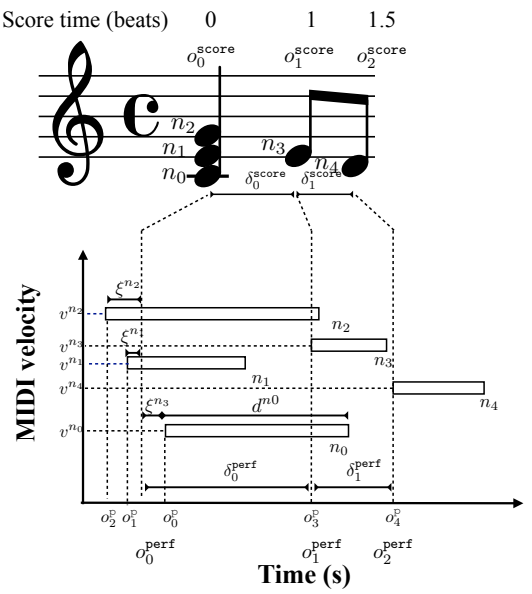

Figure 3: Excerpt of a MIDI performance (as a piano roll where the x-axis describes performance time in seconds and the y-axis is the MIDI velocity of the note) and its corresponding score, showcasing the elements for encoding/decoding an expressive performance.

time $o^{\texttt{perf}}$ (the end of the window). The input to the score follower is these windows. The full technical details of the models can be found in the Technical Appendix in the supplementary materials.

HMM Follower

The HMM-based score follower is based on the switching Kalman filter architecture, a hybrid probabilistic model which combines an HMM and a Kalman filter, whose parameters depend on the states of the HMM [Murphy, 1998]. The observed variables of this model are the performed MIDI pitch (the pitch of all of the notes inside the input 10ms windows) and $\delta^{\texttt{perf}}$, the performed IOI. The hidden variables are the set of score onset times $o^{\texttt{score}}$ plus some intermediate state for each score onset to account for possible insertions (modelled by the hidden states of the HMM), and the tempo of the performance (modelled by the Kalman filter part of the model). We can then use the forward algorithm [Rabiner and Juang, 1986; Murphy, 1998] to infer the position in the score in real-time. The design of this score follower is partially based on the probabilistic score follower discussed in [Raphael and Gu, 2009]. While this model is very reliable for relatively simple sequential input, the HMM seems to struggle with complex music, particularly music that includes cross-rhythms, ornaments or large tempo changes (see e.g., the excerpt in Fig. 4).

OLTW Follower

To address the issues with the HMM-based score follower, we use OLTW, a causal dynamic programming algorithm for aligning sequences of different lengths which incrementally aligns in real-time a streamed sequence of unknown length (in our case, the input 10 ms windows containing the MIDI performance of the soloist) to a known sequence, which we refer to as *reference* (which represents the score) [Dixon, 2005]. The performance of OLTW is controlled by 2 parameters, namely, a window size that controls how much context in the reference the algorithm considers at any given time, and a step size which controls the maximum step that OLTW is allowed to make at any given time. Following the research in real-time audio-based music alignment [Arzt and Widmer, 2015; Arzt, 2016], instead of aligning the performance directly to the score, we leverage the fact that human performers tend to play the same piece in a consistent way, and we align the input real-time performance to a previously recorded *reference performance* (ideally by the same performer), which has been aligned to the score using offline alignment methods like the one by Nakamura et al. [2015]. Furthermore, we increase the robustness of the tracking by using an ensemble of OLTW score followers, each of which aligns the input performance to a different reference performance, and we aggregate the results by computing the mean position, as proposed by Arzt and Widmer [2015].

3.2 Accompanist

The `Accompanist` module takes the score of the accompaniment part and uses the temporal alignments produced by the score follower, including the dynamics (MIDI velocity) and note duration to generate an expressive performance of the accompaniment in real-time. This module consists of two submodules, an `Encoder` that translates the input performance into a set of *expressive parameters* quantifying tempo, dynamics and articulation, and a `Decoder`, which takes these parameters to generate the performance of each of the notes in the accompaniment part.

Figure 3 shows the different elements required to encode the performance of the soloist and decode the expressive accompaniment performance. The performance of each of the notes can be encoded in 4 parameters: (1) *MIDI velocity* $v_i^{\texttt{p}}$ for capturing expressive dynamics, (2) *beat period*, defined as

$$b_i = \frac{\delta_i^{\texttt{perf}}}{\delta_i^{\texttt{score}}} \tag{1}$$

i.e., the duration of a beat in seconds (inversely proportional to tempo in beats per minute) which captures the local tempo deviations, (3) *micro-timing deviations* (denoted by ξ^i in Figure 3) which account for the onset time deviations of the notes from the global chord onset $o^{\texttt{perf}}$; and (4) the *log articulation ratio* computed as

$$a_i = \log_2 \frac{d_i^{\texttt{p}}}{d_i^{\texttt{s}} \cdot b_i} \tag{2}$$

which accounts for the ratio between performed note duration and notated duration at the current tempo).

The value of the beat period defined by Eq. 1 taken "as is" can be highly erratic in the case of real performances (see Figure 4), which typically contain large temporal fluctuations, and does not correspond with the human perception of tempo [Dixon *et al.*, 2006], which results in very unnatural (and unmusical) performances.[4] To address this is-

[4]We focus on tempo rather than musical meter, to determine when to trigger accompaniment events based on the soloist's real-time performance.

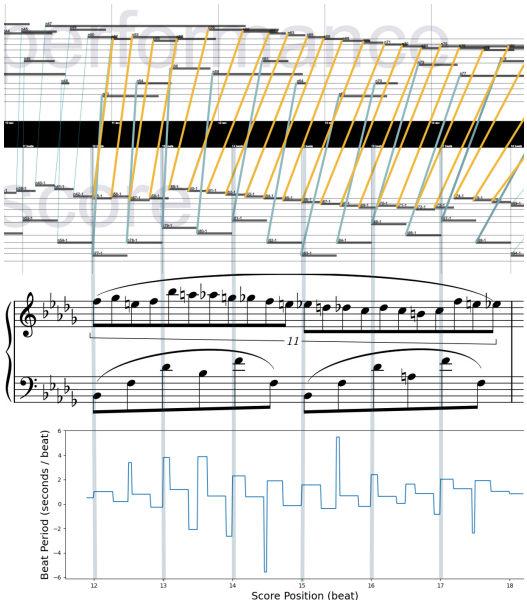

Figure 4: An excerpt of a performance of Chopin's Nocturne Op. 9 No. 1. The unusual cross-rhythm in the third measure (score in the middle) is usually played with substantial expressive freedom. Computing an IOI-based tempo based on an exact note alignment (piano rolls and connecting lines on top) leads to a jagged tempo curve (bottom), and even negative tempo values. This curve is unlikely to represent the more loose and ethereal subjective feel of the tempo in this passage.

sue, we add a tempo model inspired by the cognitive models of sensorimotor synchronization. In particular, we tested three variants, namely a *linear synchronization model* (L), which is a basic timekeeper model that adapts the tempo based on an error-correction mechanism based on the observed asynchrony [Wing and Kristofferson, 1973; Vorberg and Wing, 1996], a *linear tempo expectation model* (LTE) which expands the linear model by incorporating tempo expectations extracted from the reference performances[5] (used for the OLTW score follower) and the *joint adaptation anticipation model* (JADAM) proposed in [Van Der Steen and Keller, 2013], which includes both an error-correction term (the adaptation part) and a moving average estimate of the observed beat period (the anticipation part). Additionally, for our experiments, we use 3 baseline models that do not include asynchrony-based correction terms: a purely *reactive model* (R) which uses Eq. 1 directly, a simple *moving average* estimate of the beat period (MA) and a *Kalman filter* (K).

4 Quantitative Evaluation

We perform two experiments that focus on the quantitative evaluation of two interactive parts of our system. The first concerns the symbolic score follower, and the second evaluates the tempo models. Since, to the best of our knowledge, there are no available datasets of duet piano performances, we run both experiments using solo piano performances selected from the Magaloff [Flossmann *et al.*, 2010] and Zeilinger [Cancino-Chacón *et al.*, 2017] datasets, which consist of performances recorded on computer controlled Bösendorfer grand pianos which have been aligned to their corresponding scores. The pieces are: Nocturnes Op. 9 Nos. 1 and 2, Etude Op. 10 No. 11, Nocturne Op. 15 No. 2, the Barcarole Op. 60 by F. Chopin, and the third movement of the Sonata Op. 53 (Waldstein) by L. v. Beethoven. These pieces were selected because they contain interesting difficulties such as ornaments, trills and complex rhythms.

[5]Note that the LTE model implicitly captures musical meter.

SF	Async	$\leq$ 25ms	$\leq$ 50ms	$\leq$ 100ms
HMM	645.7	5.5	5.5	5.5
OLTW	60.6	38.0	63.3	86.7

Table 1: Alignment accuracy for the HMM- and OLTW-based score followers (SF). The first column represents the median absolute asynchrony in ms, and the remaining columns represent the percentage of asynchrony which is less than 25ms, 50ms, and 100ms accordingly, of the same pieces presented in Table 2.

4.1 Score Follower

We evaluate the ability of our score follower to follow complex performances. Since we only have one performance of each piece in the dataset, we artificially generate 5 reference performances per piece by adding zero-mean Gaussian noise with a standard deviation of 100ms to the performed onset and offset times of the notes. This process would be roughly similar (although less musical) to how musicians tend to play the same piece similarly across repeat performances [Demos *et al.*, 2016]. Table 1 compares the alignment accuracy of the HMM and OLTW score followers, and Table 2 shows how well the OLTW works for the individual pieces. To put the values in perspective, between two human instrument performers, average asynchrony varies between 30-40ms [Keller *et al.*, 2007] but also depends on the difficulty of the piece being played. These results show how OLTW can leverage information from prerecorded reference performances to improve the alignment of complex pieces (other than the Waldstein Sonata, which is arguably one of the most difficult pieces in the standard classical piano repertoire and more challenging than the other pieces in the dataset), while the HMM struggles using only information in the score (which does not include, for example, the ornament notes like trills). We also notice that the distribution of the HMM performance is bimodal, i.e., either very low or very high asynchrony.

4.2 Tempo Model

In this experiment, we evaluate the tempo models we describe above. Specifically, for each incoming onset in the soloist performance, and given the tempo computed at the previous step, the tempo model predicts a new tempo value and the next soloist onset. We then compare this prediction with the ground truth tempo and onset in the soloist performance. We run a grid search over roughly 800 parameter combinations and report values for the best parameters for each model. In case of disagreement between tempo and onset, we select the

Piece	Async	$\leq$ 25ms	$\leq$ 50ms	$\leq$ 100ms
B. Op. 53 3rd. mov.	1,813	18.4	29.3	44.4
C. Op. 9 No. 1	44	59.1	75.5	90.9
C. Op. 9 No. 2	59	42.3	62.7	87.0
C. Op. 10 No. 11	63	33.6	63.9	91.3
C. Op. 60	106	20.0	36.8	65.1

Table 2: Alignment accuracy of OLTW score follower per piece. Column 2 presents the mean asynchrony of the score followers in milliseconds. Columns 3, 4, 5 present the percentage of asynchrony which is less than 25ms, 50ms and 100ms accordingly. Hyperparameters for the OLTW score follower are 2s window size and 0.1s of step size.

Method	Onset Error (ms)	Tempo Error (ms/beat)
R	4,279.7	209.3
MA	4,271.0	203.5
L	81.9	173.1
LTE	23.3	63.3
JADAM	94.1	190.4
KT	1,154.2	177.3

Table 3: Average absolute errors in both tempo (milliseconds / beat) and onset (milliseconds) prediction for each tempo model.

best parameters based on onset prediction. Table 3 shows the results of this experiment.

The performance of the LTE model can be explained by its design advantage: this model makes use of a reference performance of the same piece to guide its estimation. In normal usage of the ACCompanion, this would be another performance the soloist recorded during their rehearsal with the system. In this experiment, this is obtained by adding Gaussian noise to the onsets of the soloist performance we consider. If the reference performance is very close to the test performance, which is the case in our experiment, the model prediction is extremely accurate. If the reference is missing, the model falls back to the linear model.

Note that a high predictive accuracy on solo piano performances does not necessarily reflect the actual goal of these models in our accompaniment setting. In our usage, the models serve two purposes: to predict the next onset and to smooth the precise, but jagged and unmusical tempo estimation based on the score follower output (see Figure 4). In this experiment, we only evaluate the first purpose, but a very high accuracy might even indicate an unsuitable algorithm that fails to meaningfully produce the human perception of tempo and follows the input too tightly. In the next section, we discuss this issue from the performer's perspective.

5 Usage and Discussion

In this section, we detail the different situations in which the ACCompanion was tested and the feedback that this generated. We then start a critical discussion about music co-performance and the challenges of human-machine collaboration in this context.

5.1 Public Live Demonstration

The ACCompanion was presented at several public venues, both to scientific and non-scientific audiences, in a two-part format. The first part consisted of a co-performance of Brahms' 5th Hungarian Dance (for four hands) by one of the authors. The system settings were manually chosen based on the preferences of the musician. In the second part, we invited people from the public to play with the system. To make the experience pleasant for people with different musical expertise without any preparation time, we selected simple pieces, in which the player only plays a short monophonic melody with the ACCompanion taking care of the accompaniment (left hand, mostly).

5.2 Musician Feedback

In lieu of a systematic user study with a sizeable number of different pianists and pieces, and as a first step towards getting a qualitative understanding of the most pressing problems that need to be addressed next, we here reproduce personal 'testimonials' by three of the authors of this paper (two of them professionally trained pianists), who have the most experience with playing with the ACCompanion. They worked on three pieces: Rondo in A major D.951 by F. Schubert, Piano Sonata K381 by W.A. Mozart, and Hungarian Dance No. 5 by J. Brahms. In Section 5.3 we will then see what insights we can distil from these personal reports.

Musician 1. Playing with the ACCompanion has been an enlightening experience. As both the main developer of the system and the person that has played with it the most, I am very aware of its shortcomings, but I am also very pleased (and perhaps, dare I say proud) of each (small) breakthrough. Playing with the system has broadened my perspectives on what aspects are important for ensemble performance, in particular it has made me realize the importance of understanding and communicating shared expressive intentions between musical partners. Every time that I play duets with other humans, I find myself thinking of what the ACCompanion would do, and it has made me more aware of how difficult it is to play with someone else, and how easily can things go wrong when the expressive intentions for the piece have not been discussed (e.g., when sight-reading unfamiliar music). While the ACCompanion is still (very) far from being able to accompany a performance of Rachmaninoff's 3rd Piano Concerto (my personal goal for the system), it can be a rewarding experience. For example, in the performance of Brahms' piece[6], it is very nice how the system slows down at the fermata at the end of part B, and then just starts playing on time afterwards. In those cases, I can (almost) forget that I'm playing with an artificial partner and just focus on the music that I'm playing.

Musician 2. Playing with the ACCompanion was a fun, albeit somewhat troublesome experience. My piece starts with an eighth-note chord, after which I start playing a sequence of sixteenth notes, to which the system then responds with a similar sequence before we play along jointly. In the initial trials, the system would run away immediately by starting off with a crazy fast tempo. Even if I jumped in at the

[6] https://youtu.be/Wtxcqp-sQ_4

correct position (i.e., current system score position), it was not possible anymore to play jointly together, as the accompaniment would not slow down sufficiently. After a lot of trial and error, we eventually figured out this seemingly random behaviour was caused by the initial chord and the (what the ACCompanion perceived as) "pause" before the sixteenth notes sequence, which caused the score follower to frantically align notes where there were none, and the tempo model to adjust the global tempo accordingly. After we cut this initial chord (i.e., starting the piece at my sequence directly), we were playing jointly and in the same (global) tempo, and the ACCompanion then was able to respond effortlessly to my playing style, especially in terms of (micro-)timing and dynamics. Overall, I had the impression that the system was *re*acting more than it was acting on its own, similar to when rehearsing for the first time with a human ensemble partner who is not yet too familiar with ensemble playing.

Musician 3. Playing with the ACCompanion was an encounter of the third kind, and a rather stressful one, for me. The moment that epitomizes it all is the very beginning of a performance: you play the first note, and you just *hope* that the ACCompanion will join in, in the right tempo. It (almost) always does, in the end, but the very fact that you have to worry about it points to the central problem: a complete lack of natural communication and trust. For the next few bars, you are busy being relieved that the start 'worked', and then, for the rest of the piece, you very consciously focus on pulling it along, tricking it into speeding up or slowing down (which might or might not be necessary; but you just don't trust it), probing how much spontaneous change it can handle – in short: you are constantly focused on *it*. The result of my efforts can be seen in this video[7] (with apologies to Franz Schubert) – my playing is unrelaxed, unnatural throughout. So: this research teaches us as much about what's missing as it does about probabilistic tempo modeling or performance prediction. To me, the ultimate basis is *trust*: in each other's musical understanding, but also in my partner's ability and readiness to sense when I struggle and support me.

5.3 On Human-Machine Expressive Collaboration

Using the framework of togetherness proposed by Bishop [2023], the extent of expressive musical collaboration can be placed on a spectrum ranging from mere awareness of others (i.e., the lower bound of interaction) to the experiential process of cognitive and emotional alignment. While collaborative musical interaction between humans can be studied from this phenomenological standpoint, the same approach cannot be taken when investigating human-machine collaboration, as systems inherently do not possess any internal state and are not capable of perceiving music (or sound, for that matter) other than in the way they are designed and built to do. Indeed, our musical experiences with the system suggest that the interaction the ACCompanion has with the soloist is biased towards following and, for the most part, our system misses out on making its own decision. For example, we observe that when the system "believes" that the human player is slowing down (which may not be the case), it slows down in response, which causes the human player to slow down as well, which in turn causes the system to slow down even more. The system reaction, in this case, is contrary to what a human accompanist would do, i.e., to try to keep the tempo stable, and results in a decreased sense of togetherness.

In Section 2 we reported the subtasks existing accompaniment systems focus on: note detection (for audio systems), score following, and expressive accompaniment generation. We believe that a missing fourth point should be added to this list: *modeling the feedback loop with the human partner*. This means that the system needs to "understand" when to be more *reactive* and follow the human player, and when more *proactive* and lead the performance. Some system upgrades that would help in this direction are: further study of tempo models, the usage of the long-term musical structure of the piece, and the introduction of visual cues. The latter can be considered in the music co-performing context, as useful as non-verbal communication is in the speech context. For example, let us consider the challenging situation of a chord after a long rest. Musicians would strongly rely on visual cues, such as breathing, nod, and hand gestures, to synchronise. Given our previous considerations on leading and following, we would need visual cues for both the system (e.g., signals from body sensors or a camera), and the player (e.g., video or haptic feedback on the state of the tempo model).

6 Conclusion

This paper introduces the ACCompanion, an automatic system capable of accompanying a human soloist performing a given musical score. It can work with a variety of different inputs, as long as the full musical score is given in a symbolic format, the soloist performance can be encoded in MIDI data, and a system that translates MIDI outputs to sound is available. We describe the two tasks our accompanist must perform: online score following and automatic accompaniment generation, and we approach their interaction with cognitively plausible models inspired by research on sensorimotor synchronization. We perform two quantitative experiments to evaluate critical parts of our system, test it at multiple public live demonstration, and evaluate it personally from a musician's perspective. Based on the experimental results, audience feedback and our own experience, we start a critical discussion on the still open challenges for accompaniment systems to be precise and reliable musical partners

Future development of the ACCompanion will go in the direction of modeling the feedback loop with the human partner, for the system to understand when to follow the soloist and when to lead the performance. Furthermore, we plan a systematic user evaluation of the entire system in different configurations, based e.g., in the work by Zhou et al. [2023]. Moreover, we are working on a new end-to-end accompanist part, that would produce an expressive accompaniment conditioned to the soloist performance in a single step, with the advantage of sharing the information about the two tasks, to improve both. Finally, we plan on introducing visual cues for both the soloist, in the form of visual feedback on the computed tempo, and the accompaniment system, in the form of signals from body sensors or a camera.

[7] https://youtu.be/qEocywdruco

Ethical Statement

There are no ethical issues.

Acknowledgments

We want to thank Laura Bishop for her contributing recordings to train the ACCompanion. This work is supported by the European Research Council (ERC) under the EU's Horizon 2020 research & innovation programme, grant agreement No. 10101937 ("Whither Music?").

References

[Agrawal and Dixon, 2019] Ruchit Agrawal and Simon Dixon. A Hybrid Approach to Audio-To-Score Alignment. In *Proceedings of the International Conference on International Conference on Machine Learning*, 2019.

[Arzt and Lattner, 2018] Andreas Arzt and Stefan Lattner. Audio-To-Score Alignment using Transposition-Invariant Features. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[Arzt and Widmer, 2015] Andreas Arzt and Gerhard Widmer. Real-Time Music Tracking Using Multiple Performances as a Reference. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[Arzt, 2016] Andreas Arzt. *Flexible and Robust Music Tracking*. PhD thesis, Johannes Kepler University Linz, Austria, 2016.

[Bishop, 2023] Laura Bishop. Focus of attention affects togetherness experiences and body interactivity in piano duos. *Psychology of Aesthetics, Creativity, and the Arts*, 2023.

[Cancino-Chacón *et al.*, 2017] Carlos Eduardo Cancino-Chacón, Thassilo Gadermaier, Gerhard Widmer, and Maarten Grachten. An Evaluation of Linear and Nonlinear Models of Expressive Dynamics in Classical Piano and Symphonic Music. *Machine Learning*, 106(6):887–909, 2017.

[Cancino-Chacón *et al.*, 2018] Carlos E Cancino-Chacón, Maarten Grachten, Werner Goebl, and Gerhard Widmer. Computational Models of Expressive Music Performance: A Comprehensive and Critical Review. *Frontiers in Digital Humanities*, 5:25, 2018.

[Cancino-Chacón *et al.*, 2022] Carlos Eduardo Cancino-Chacón, Silvan David Peter, Emmanouil Karystinaios, Francesco Foscarin, Maarten Grachten, and Gerhard Widmer. Partitura: A Python Package for Symbolic Music Processing. In *Proceedings of the Music Encoding Conference (MEC2022)*, Halifax, Canada, 2022.

[Chen *et al.*, 2014] Chun-Ta Chen, Jyh-Shing Roger Jang, and Wenshan Liou. Improved Score-Performance Alignment Algorithms on Polyphonic Music. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1365–1369. IEEE, 2014.

[Cont, 2008] Arshia Cont. ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 33–40, 2008.

[Dannenberg, 1984] Roger B Dannenberg. An On-Line Algorithm for Real-Time Accompaniment. In *Proceedings of the International Computer Music Conference (ICMC)*, volume 84, pages 193–198, 1984.

[Dannenberg, 1987] Roger B Dannenberg. Following an Improvisation in Real-Time. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 241–248, 1987.

[Demos *et al.*, 2016] Alexander P Demos, Tânia Lisboa, and Roger Chaffin. Flexibility of expressive timing in repeated musical performances. *Frontiers in Psychology*, 7:1490, 2016.

[Dixon *et al.*, 2006] Simon Dixon, Werner Goebl, and Emilios Cambouropoulos. Perceptual Smoothness of Tempo in Expressively Performed Music. *Music Perception*, 23(3):195–214, 2006.

[Dixon, 2005] Simon Dixon. An On-Line Time Warping Algorithm for Tracking Musical Performances. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1727–1728, 2005.

[Duan and Pardo, 2011] Zhiyao Duan and Bryan Pardo. A State Space Model for Online Polyphonic Audio-Score Alignment. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 197–200. IEEE, 2011.

[Flossmann *et al.*, 2010] Sebastian Flossmann, Werner Goebl, Maarten Grachten, Bernhard Niedermayer, and Gerhard Widmer. The Magaloff Project: An Interim Report. *Journal of New Music Research*, 39(4):363–377, 2010.

[Friberg *et al.*, 2006] Anders Friberg, Roberto Bresin, and Johan Sundberg. Overview of the KTH Rule System for Musical Performance. *Advances in Cognitive Psychology*, 2(2):145, 2006.

[Gingras and McAdams, 2011] Bruno Gingras and Stephen McAdams. Improved Score-Performance Matching using Both Structural and Temporal Information from MIDI Recordings. *Journal of New Music Research*, 40(1):43–57, 2011.

[Jeong *et al.*, 2019] Dasaem Jeong, Taegyun Kwon, Yoojin Kim, Kyogu Lee, and Juhan Nam. VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 908–915, 2019.

[Keller *et al.*, 2007] Peter E Keller, Günther Knoblich, and Bruno H Repp. Pianists duet better when they play with themselves: on the possible role of action simulation in synchronization. *Consciousness and cognition*, 16(1):102–111, 2007.

[Loehr *et al.*, 2011] Janeen D Loehr, Edward W Large, and Caroline Palmer. Temporal Ccoordination and Adaptation to Rate Change in Music Performance. *Journal of Experimental Psychology: Human Perception and Performance*, 37(4):1292, 2011.

[Maezawa *et al.*, 2018] Akira Maezawa, Kazuhiko Yamamoto, and Takuya Fujishima. Rendering Music Performance with Interpretation Variations using Conditional Variational RNN. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[Mates, 1994] Jiří Mates. A Model of Synchronization of Motor Acts to a Stimulus Sequence: I. Timing and Error Corrections. *Biological Cybernetics*, 70(5):463–473, 1994.

[Murphy, 1998] Kevin P Murphy. Switching Kalman Filters. Technical Report 98-10, Compaq Cambridge Research Lab, August 1998.

[Nakamura *et al.*, 2014] Eita Nakamura, Nobutaka Ono, Yasuyuki Saito, and Shigeki Sagayama. Merged-Output Hidden Markov Model for Score Following of MIDI Performance with Ornaments, Desynchronized Voices, Repeats and Skips. In *International Conference on Mathematics and Computing*, 2014.

[Nakamura *et al.*, 2015] Tomohiko Nakamura, Eita Nakamura, and Shigeki Sagayama. Real-Time Audio-To-Score Alignment of Music Performances containing Errors and Arbitrary Repeats and Skips. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(2):329–339, 2015.

[Nakamura *et al.*, 2017] Eita Nakamura, Kazuyoshi Yoshii, and Haruhiro Katayose. Performance Error Detection and Post-Processing for Fast and Accurate Symbolic Music Alignment. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 347–353, 2017.

[Peter *et al.*, in press] Silvan Peter, Carlos Eduardo Cancino-Chacón, Emmanouil Karystinaios, Francesco Foscarin, Andrew McLeod, and Gerhard Widmer. Automatic Note-Level Score-To-Performance Alignments in the ASAP Dataset. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, in press.

[Rabiner and Juang, 1986] Lawrence Rabiner and Biinghwang Juang. An Introduction to Hidden Markov Models. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 3(1):4–16, 1986.

[Raphael and Gu, 2009] Christopher Raphael and Yupeng Gu. Orchestral Accompaniment for a Reproducing Piano. In *International Conference on Mathematics and Computing*, 2009.

[Raphael, 2010] Christopher Raphael. Music Plus One and Machine Learning. In *Proceedings of the International Conference on International Conference on Machine Learning*, pages 21–28, 2010.

[Repp, 2006] Bruno Hermann Repp. Musical Synchronization. *Music, Motor Control, and the Brain*, pages 55–76, 2006.

[Van Der Steen and Keller, 2013] Maria Christine Van Der Steen and Peter E Keller. The ADaptation and Anticipation model (ADAM) of Sensorimotor Synchronization. *Frontiers in human neuroscience*, 7:253, 2013.

[Vercoe, 1984] Barry Vercoe. The Synthetic Performer in the Context of Live Performance. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 199–200, 1984.

[Vorberg and Wing, 1996] Dirk Vorberg and Alan Wing. Modeling Variability and Dependence in Timing. In *Handbook of Perception and Action*, volume 2, pages 181–262. Elsevier, 1996.

[Widmer *et al.*, 2009] Gerhard Widmer, Sebastian Flossmann, and Maarten Grachten. YQX plays Chopin. *AI magazine*, 30(3):35–35, 2009.

[Wing and Kristofferson, 1973] Alan M Wing and Alfred B Kristofferson. Response Delays and the Timing of Discrete Motor r Responses. *Perception & Psychophysics*, 14(1):5–12, 1973.

[Xia and Dannenberg, 2015] Guangyu Xia and Roger B Dannenberg. Duet Interaction: Learning Musicianship for Automatic Accompaniment. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 259–264, 2015.

[Xia *et al.*, 2015] Guangyu Xia, Yun Wang, Roger B Dannenberg, and Geoffrey Gordon. Spectral Learning for Expressive Interactive Ensemble Music Performance. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 816–822, 2015.

[Zhou *et al.*, 2023] Zijun Zhou, Justin Christensen, Jorden A. Cummings, and Janeen D. Loehr. Not just in sync: Relations between partners' actions influence the sense of joint agency during joint action. *Consciousness and Cognition*, 111:103521, 2023.

A Technical Appendix: Sensorimotor Synchronization Models

This Appendix follows the notation introduced in Section 3 (in particular Figure 3), but we drop the superscript p of performed onsets to unclutter notation. In the following description of the tempo models, we denote the observed performed onset time of the n-th onset in the score as o_n and the onset time predicted by the synchronization models as $\hat{o}_n$. The asynchrony between these onsets is denoted as $A_n = \hat{o}_n - o_n$ and the observed beat period is given as $\tau_n = \frac{\delta_n^{\mathrm{perf}}}{\delta_n^{\mathrm{score}}}$, with τ_0 being the initial tempo set by the performer as a hyper parameter.

A.1 Reactive Sync Model (R)

The next accompaniment onset is "estimated" at the same time as the last observed performed onset, making this model purely reactive.

$$\hat{o}_{n+1} = o_n + b_n \delta_n^{\mathrm{score}} \tag{3}$$
$$b_{n+1} = \tau_n \tag{4}$$

A.2 Moving Average Sync Model (MA)

This model is very similar to the first baseline, except for its estimation of the global tempo, which uses a weighted average of the last and current tempo estimate.

$$\hat{o}_{n+1} = o_n + b_n \delta_n^{\mathrm{score}} \tag{5}$$
$$b_{n+1} = \eta_{\mathrm{MA}} b_n + (1 - \eta_{\mathrm{MA}})\tau_n \tag{6}$$

where η_{MA} is a constant parameter.

A.3 Linear SMS Model (L)

Onset and beat period estimates o_{n+1} and b_{n+1} are computed as follows:

$$\hat{o}_{n+1} = \hat{o}_n + b_n \delta_n^{\mathrm{score}} - \eta^o A_n \tag{7}$$
$$b_{n+1} = \begin{cases} b_n - \eta^b A_n, & \text{if } A_n < 0, \\ b_n - 2\eta^b A_n, & \text{else} \end{cases} \tag{8}$$

where η^o and η^b are the learning rates for the onset and beat period, respectively.

A.4 Linear Tempo Expectation Model (LTE)

This model is similar to the previous one, but includes an anticipation (expectation) term in the computation of the beat period:

$$\hat{o}_{n+1} = \hat{o}_n + b_n \delta_n^{\mathrm{score}} - \eta^o A_n \tag{9}$$
$$b_{n+1} = \phi(o_{n+1}^{\mathrm{score}}) - \eta^b A_n, \tag{10}$$

where $\phi(o_n^{\mathrm{score}})$ is a function that computes an estimate of the beat period at score onset time o_n^{score} based on the tempo of the reference performance(s) and η^o and η^b are constant parameters that represent the learning rates for the onset and beat period, respectively.

A.5 Joint Adaptation Anticipation Model (JADAM)

This model includes both an error-correction term (the adaptation part) and a moving average estimate of the observed beat period (the anticipation part).

1. Adaptation Module

$$\hat{o}_{n+1}^{\mathrm{ad}} = \hat{o}_n + b_n \delta_n^{\mathrm{score}} - \eta_{\mathrm{J}}^o A_n \tag{11}$$
$$b_{n+1} = b_n - \eta_{\mathrm{J}}^b A_n \tag{12}$$

2. Anticipation Module

$$\hat{\tau}_n = \eta_{\mathrm{J}}^b (2\tau_n - \tau_{n-1}) + (1 - \eta_{\mathrm{J}}^b)\tau_n \tag{13}$$
$$\hat{o}_{n+1}^{\mathrm{an}} = o_n + \hat{\tau}_n \delta_n^{\mathrm{score}} \tag{14}$$

3. Joint Module

$$\hat{A}_n = \hat{o}_{n+1}^{\mathrm{ad}} - \hat{o}_{n+1}^{\mathrm{an}} \tag{15}$$
$$\hat{o}_{n+1} = \hat{o}_{n+1}^{\mathrm{an}} - \eta_{\mathrm{J}}^a \hat{A}_n \tag{16}$$

Parameters η_{J}^o and η_{J}^a and η_{J}^b are learning rates for the prediction of the onset of the adaptation and joint modules and the learning rate for the beat period, respectively.

A.6 Kalman Tempo Model (KT)

The observed variable of the Kalman filter is the performed onset time, and the beat period is the latent variable. The updates are computed as follows:

$$\hat{b}_n = \alpha_{\mathrm{K}} b_n \tag{17}$$
$$v_n = \gamma_{\mathrm{K}}^2 \hat{v}_n + \beta_{\mathrm{K}} \tag{18}$$
$$\hat{A}_n = \delta_n^{\mathrm{perf}} - \hat{b}_n \delta_n^{\mathrm{score}} \tag{19}$$
$$\kappa_n = \frac{v_n \delta_n^{\mathrm{score}}}{v_n \delta_n^{\mathrm{score}^2} + \lambda_{\mathrm{K}}} \tag{20}$$
$$b_{n+1} = \hat{b}_n + \kappa_n \hat{A}_n \tag{21}$$
$$\hat{v}_{n+1} = (1 - \kappa_n \delta_n^{\mathrm{score}}) v_n \tag{22}$$
$$\hat{o}_{n+1} = \hat{o}_n + b_{n+1} \delta_n^{\mathrm{score}} \tag{23}$$

where α_{K}, β_{K}, γ_{K} and λ_{K} are the parameters of the model.