“Video Accompaniment”: Synchronous Live Playback for Score-Aligned Animation

Kaitlin Pet
Indiana University
kpet@iu.edu

Nikki Pet
Yale School of Music
nikki.pet@yale.edu

Christopher Raphael
Indiana University
craphael@indiana.edu

ABSTRACT

We developed a “video accompaniment” system capable of closely aligning a pre-made music video to a live, tempo-varying musical performance. Traditionally, “tight” video-to-audio synchrony is only capable with a musician-restricting system like a click track, or making a human operator responsible for the timing of visual content. Our system automatically aligns the video to a live music performance. It uses the Informatics Philharmonic automatic accompaniment software to 1) follow a musician’s score position in real time and 2) predict when the next score position will occur. These position predictions are used to stretch the video such that animated gestures align with their musical counterparts. We worked with clarinetist and animator Nikki Pet to adapt two musical works by composer Joan Tower to this new medium. These works were performed at multiple venues with our video accompaniment system. This paper will describe both the design details of the video accompaniment system and our performance and development experience. We will end by discussing the artistic ramifications and future improvements for this technology.

1. INTRODUCTION

Often, innovation in technology stems from specific real-world needs. The artistic drive behind this project was the desire of Nikki Pet (paper author and sister to author Kaitlin Pet) to use her classical music accompaniment animation in live performance. During the COVID-19 lockdown, Nikki produced music videos where she augmented classical music recordings with highly-synchronized animation (for examples, see [1,2]). Though time-consuming, the process of producing such videos is relatively technologically straightforward – Nikki layered short snippets of animation in a video editing software so specific animated motions in the video synchronized with corresponding musical gestures in the pre-recorded audio.

Nikki wanted to incorporate this type of highly synchronized animation in live concert, in addition to videos produced offline. However, current methods for performing live alongside animation are 1) not user-friendly and 2) poorly suited to a performance practice involving fluid tempo changes. If the video is a “fixed” media, the musician is required adjust their ideal tempo and interpretation to stay with the video instead of following their own temporal interpretation. For example, in applications such as silent movie accompaniment and “Movies at the Symphony” concerts, players often have an in-ear microphone with a “click track” specifying the specific tempo and timing they need to stay in line with the video.

We thus wanted to create a system for live animation without these restrictions, a system where animation created to synchronize with music in a specific way could be “stretched” in real time to align with a live player. This system should make animation “follow” the soloist, instead of requiring the musician to constantly match their performance interpretation to a static video.

To implement this video accompaniment system for classical music, we must both track the musician’s score position in real time, and modulate frame rate in a way that synchronizes with the musician. Score following in the context of classical music audio accompaniment has been studied for many years since being developed by Dannenberg and Vercoe in the mid 1980s [3, 4]. Score following exploits a key trait in classical music: performers will often follow a pre-determined set of notes and rhythms instead of creating original musical content on-the-spot. This pre-knowledge of a performer’s action can be used to both track the soloist’s position in the score and execute score-aligned actions. Often, these actions are discreet, such as MIDI accompaniment generation or performance-adjacent utilities like page turning [5–9]. Another class of automatic audio accompaniments using phase vocoding to make a pre-recorded audio track align with a live soloist [10, 11]. One such vocoder-based automatic accompaniment system, the Informatics Philharmonic by Christopher Raphael [10], serves as technological base for our current project.

We first developed a system that adapts Informatics Philharmonic to a more general system able to control arbitrary audio or visual content specifiable by the Max/MSP graphical programming language [12]. This system was first proposed by our previous work in [13]. Instead of controlling a pre-recorded audio track, the Informatics Philharmonic’s control data is sent to a Max/MSP patch. The patch can then utilize the control information for arbitrary creative purposes, including score-aligned, realtime generation of audio and video. Using this as a foundation, we developed a system to dynamically change the video playback rate of a ‘score-synchronous’ video to align with live music performance. During a performance, the musician uses Informatics Philharmonic to follow their position in the score and predict future note onset times. The Max/MSP patch then translates those note onset predictions into live synchronous animation.

In tandem to our development of the video accompaniment system, Nikki created two multimedia works designed for synchronous live accompaniment. Nikki reached out to classical composer Joan Tower about augmenting Tower's piece *Wings* (for solo clarinet) with animation. Tower then requested Nikki also adapt another one of her pieces: *Fanfare for the Uncommon Woman No. 5* arranged for clarinet quartet. Nikki created two full-length animation videos, one evoking birds for *Wings* and a more abstract animation inspired by female musicians for *Fanfare for the Uncommon Woman No. 5*.

We have thus far given six performances using this video accompaniment technology. For each performance, Nikki was either the solo performer or a member of a performance group. We quantitatively evaluate the accuracy of one of these performances, giving numerical insight into video alignment accuracy and provide a benchmark for future research. We then qualitatively evaluate these performances and the system as a whole. For qualitative evaluation, we use both audience response and feedback from Nikki about her experience designing animation and performing music with the system. We will discuss both successful aspects of the performances, and ways we can improve future iterations of the technology. We hope the advantages and limitations we observed "in the field" will be informative for future artistic endeavours using similar technologies.

2. SYSTEM DESIGN

For any type of score-based multimedia accompaniment, one must consider how to translate information yielded by the score follower to synchronous actions or gestures in the multimedia. We will focus on the use case where a synchronous, **pre-specified** multimedia accompaniment is desired – i.e. the live part is meant to align with pre-arranged events and gestures in a specific way. We have observed two main types of strategies in the literature and existing score-based accompaniment applications.

The first strategy is *responsive*: when the system is sufficiently certain a note has been played, a signal is triggered to the multimedia generator. This reactive strategy has been used with systems such as PHENICX [9] and Orchestral Performance Companion [14], which demonstrate either artistic or informative visual information to an audience at pre-defined positions in a classical music performance. Antescofo [15, 16] allows one to customize this type of multimedia synchronization in Max/MSP and Pure Data – one can specify the release of a previously specified message when a specific note in the loaded score is recognized. This message can then be used to trigger any downstream event specifiable in Max/MSP, including fixed or generative visual content.

While this reactive strategy is capable of supporting interesting artistic applications, it is sub-optimal for aligning continuous media like video in a smooth and synchronous way. Consider a hypothetical video showing a person throwing then catching a ball. A score-synchronized music performance may require note n in the score to align with the ball-throwing action, and note $n + 1$ in the score to align with the ball-catching action. For the ball's movement to appear smooth and realistic, the video frame rate between the onset times of n and $n+1$ should be relatively constant, thus minimally deforming the ball's original acceleration pattern. This ideal constant frame rate can be computed by:

$$R_n = \frac{\Delta f_n}{\Delta t_n} \tag{1}$$

Where Δt_n is the time between note onset n and note onset $n + 1$ and Δf_n is the number of video frames between when the ball is thrown and when it is caught. The video should ideally start playing at rate R_n as soon as note n is played – this way the ball's movement will appear consistent with the laws of physics and its temporal trajectory will not be deformed. However, this optimal rate R_n cannot be computed without knowledge (or an estimate) of what time note $n + 1$ will be played by the musician.

We thus need a mechanism of *predicting* the position of future note onsets before they happen. In this strategy, information gleaned from previously recognized notes in the score are used to extrapolate the position of the live player's future note placement. Many systems, including Antescofo and other MIDI-based automatic accompaniment systems, provide an estimate of the musician's current tempo – one can use this tempo to extrapolate the likely position of future note onsets. This strategy has also been used in the context of robotics, for example controlling the fingers of a robot accompanist/collaborator [17].

At a high level, our multimedia alignment system consists of three parts: a score follower, a prediction engine, and a video playback rate modulator. The Informatics Philharmonic [10] functions as the score follower and prediction engine. Informatics Philharmonic maps live performance audio of a pre-determined piece into a sequence of notes in the score, then provides the predicted time of the musician's next note onset. This predicted onset time is then sent to a Max/MSP patch, which controls the playback rate of video in real time. To generate animation that aligns with a live soloist, the Max/MSP patch bases its rate calculation on points of synchrony specified in an arbitrary score-aligned video. In the next sections, we discuss in more detail how these components function and work together.

2.1 Informatics Philharmonic

The Informatics Philharmonic was created as a classical music automatic accompaniment system. Its score follower and accompaniment scheduler are used to control video accompaniment. During rehearsal or performance, the Informatics Philharmonic read in the musician's audio and analyzes it using a Hidden Markov Model(HMM)-based online score follower to determine the soloist's trajectory through the score. The Informatics Philharmonic uses these identified note onset times to schedule future accompaniment events. Specifically, it uses a Kalman filter-like mechanism to simultaneously predict the position of the $t+1^{th}$ note in the score and the tempo of the $t+1^{th}$ note in the score:

$$s_{n+1} = s_n + \sigma_n \tag{2}$$

$$t_{n+1} = t_n + l_n s_n + \tau_n \tag{3}$$

Where s_n is the tempo at note n, t_n is the time of note n, l_n is the length (in beats) of note n, and σ_n and τ_n are normally distributed "trainable" noise variables. These noise terms allow one to learn a musician's timing tendencies from previous takes – consistent musical tendencies such as tempo changes or agogic accents can be anticipated by the system, increasing prediction accuracy.

This model is built for a classical music accompaniment scenario, a "two-way" system where the soloist influences the audio accompaniment, but the accompaniment also influences the soloist. In this framing, solo and accompaniment parts are modeled as part of a single "combined interpretation" – observed solo note times from the score follower and the times of previously played notes in the accompaniment track are both used in scheduling. This promotes steadiness and internal consistency in the accompaniment part playback, and is intended to allow for a more realistic collaborative experience for musicians accustomed to playing with human accompanists. However, in the video accompaniment use case, we assumed a "one way" information flow where the video responds to the live musician but the musician does not respond to the video. We considered video accompaniment to be a less collaborative medium, with the soloist "controlling" the video instead of "playing with" the video. Thus, we wanted Informatics Philharmonic to simply yield the predicted time of the next solo note, so the accompanying video could match it as closely as possible. This could largely be achieved by setting the accompaniment score in unison to the live musician's part. Thus,the scheduled time of note $n+1$ in the accompaniment corresponded to the predicted time of note $n+1$ in the live musician's part. Note that this "one direction" assumption had unexpected artistic and practical consequences, which will be discussed in Section 3.4.

In the original audio-accompaniment function for Informatics Philharmonic, these scheduled times of future note onsets are used to speed up or slow down the playback rate of a pre-recorded accompaniment track so it retains the correct rhythmic relationship with the soloist. Usually, the accompaniment track's instantaneous rate of change, $\frac{dp}{dt}$, is calculated by:

$$\frac{dp}{dt} = \frac{\Delta p}{\Delta t} \tag{4}$$

where Δp is the distance from the accompaniment track's current playback position and the position of the next parsed accompaniment note, and Δt is the time difference between the current time and the next note's predicted onset time. This is very similar to what we described Equation 1 as the ideal video alignment playback rate, R_n.

However, there are certain cases where a soloist's pending note onset cannot be effectively predicted from their past performance. For example, perhaps the soloist needs to pause and breathe at a certain point in the score, temporarily stopping the musical flow. To address this type of situation, Informatics Philharmonic has a "cueing" mechanism which temporarily switches to the responsive framework described in the beginning of Section 2. Instead of predicting these note onsets ahead of time, Informatics Philharmonic waits until the cued onset is heard before scheduling the corresponding accompaniment event. This responsive strategy makes assigning a phase vocoder rate more complicated. Immediately preceding a cue, the vocoder rate $\frac{dp}{dt}$ changes such that:

$$\frac{dp}{dt} = c(p_0 - p) \tag{5}$$

where c is a scaling constant, p is the current playback position of the accompaniment recording, and p_0 is the position of the pending cued note in the accompaniment recording. Using this framing, the audio gets slower and slower but never reaches the position in the accompaniment associated with the cue. After the cued note is heard, the phase vocoder is set to a predetermined rate until the next note's position in predicted.

We implemented a UDP port in Informatics Philharmonic to send information about the pending solo note's onset times to a Max/MSP patch via OSC protocol. In cases where the next onset time can be reasonably predicted, we send the scheduled time of the next note onset. Otherwise, a message is sent expressing the next note's onset cannot be reliably predicted. This information is used by the Max/MSP patch described in Section 2.2 to compute video frame rate and control video playback.

2.2 Controlling Video Playback in Real Time

Before designing the Max/MSP video control patch, we needed an operationalized definition of what it means for video events to be "in sync" with notes in a classical music score. In a previous example, we described a note n occurring right as video "shows a person throwing a ball" – but what does this mean from an implementation perspective? We chose to use the concept of animation "keyframes" in order to define points of coincidence. We define a keyframe as a frame index in the animation that should exactly align with a certain note. "Perfect" synchrony thus means that the live musician plays note n at the exact same time that note n's corresponding keyframe is shown during a live performance.

Thus for each of the synchronized animations we created, a mapping of note onsets to their associated keyframes are stored in a **measure-to-frame dictionary**. The **measure-to-frame dictionary** is then used by the Max/MSP patch during live performance for video rate calculation, described in more detail below.

We implemented realtime video control using the `jit.movie` object, which displays and allows for fine-grain control of pre-loaded video files. We found the most reliable way to align video was to continually compute the position (in frames) the video should be at during the performance, then immediately tell `jit.movie` to jump to the correct position.

Thus, our job becomes calculating the current frame index of the animation needed to align with the live musician. To achieve this, we translate information about the pending solo note during a live performance into an appropriate instantaneous video playback rate. This playback rate is used to compute the current video position until new information from the Informatics Philharmonic causes a revised rate to be computed.

The current video rate is computed in two ways, depending on whether the Informatics Philharmonic schedules using a predictive or responsive strategy (described in 2.1):

- **Predictive Control Flow**

  1. The Max/MSP patch receives a message from Informatics Philharmonic in the form of "Score position X should occur at time T_x".
  2. We look up the video keyframe K_x that should coincide with score position X in the **measure-to-frame dictionary**.
  3. We calculate the rate R_x, in frames/second, needed to reach keyframe K_x by time T_x where

  $$R_x = \frac{\Delta F}{\Delta T} \qquad (6)$$

  where ΔF is the distance between the current video position and keyframe K_x, and ΔT is the amount of time from now until T_x. Note that unlike the rate described by Equation 1, this rate is initialized when the prediction message is received, rather than right when the previous score position occurs.

  Note that this method places no limitations on a 'valid' rate – if the current video position exceeds the position of the next specified keyframe, the video will play backwards. We have observed backward playback as a form of "course correction" if the previous specified rate was too fast, see example here. This freedom to move forward and backward through the piece can create challenges when quantitatively evaluating accuracy, which will be discussed in Section 3.1.

- **Responsive Control Flow**

  1. The Max/MSP patch receives a message from Informatics Philharmonic in the form "Score position X is a cuepoint". This means that the live musician is responsible for signalling the onset of score position X, thus the time associated with X is not known in advance.
  2. We look up the video keyframe K_x corresponds to score position X in the **measure-to-frame dictionary**.
  3. We do not update the playrate rate R. If the cued note is heard before keyframe K_x is reached, the animation jumps ahead to K_x to align with the soloist. If the cued note is heard after K_x is reached, the video "pauses" until new messages are received. See a video demonstration of this pausing behavior here.

  Note that this strategy to pause or jump forward at cues was chosen based mainly on implementation ease and anticipated appropriateness for most artistic purposes. We will discuss the effectiveness of this strategy as well as response from the performers and audience in Section 3.4.

2.3 Creating the Score-matched animation

We wanted the two components above to form a generalizable system compatible with an arbitrary score-matched animation. We thus developed a process for creating score-matched animations which did not require the animator (Nikki) to be aware of implementation details of the synchronization system.

To create score-matched animation, the animator only requires a tempo-marked score in computer-readable format, and a musical rendering of the file. In our case, we created a MIDI files and MIDI-rendered clarinet audio from *Wings* and *Fanfare For the Uncommon Woman No. 5*. Nikki then used this MIDI-rendered clarinet audio as a "guide audio track" to create aligned animations in Final Cut Pro.

We then used an automated pipeline based on the `music21` package [18] to extract tempo and note length information from the MIDI file. This was used along with the animation video frame rate to compute the position in the video (in frames) corresponding to each note in the score, forming the **measure-to-frame dictionary** used in Section 2.2 to look up each score-synchronous keyframe during performance.

3. RESULTS

We gave six live concerts using this live animation alignment system. Nikki Pet performed in all concerts with *Wings*, and Nikki and three other professional clarinetists performed in the concert for *Fanfare for the Uncommon Woman No. 5*. Audience members for these concerts ranged the gamut from trained music professionals, music students, and elementary schoolchildren. All concerts used a Blue Yeti microphone for audio input, and either a portable projector or a venue-specific projector to display the animation.

We will first provide objective alignment accuracy metrics of a single performance from Nov 29, 2022 using the most recent iteration of the technology. We will then synthesize qualitative feedback from performers and audience members on aspects where the system was more or less successful, as well as reflecting on the efficacy of the design pipeline. We link here the live performance of *Wings* from November 29, 2022, and here the performance of *Fanfare for the Uncommon Woman No. 5* from December 4, 2022. Readers can watch these performances to form their own judgements about the system's perceptual accuracy. These performances includes sections of very fast clarinet playing where each clarinet note is aligned with a specific animated gesture. For example, here and here show instances in *Wings* where a node on a graph flashes to match every clarinet note.

3.1 Quantitative Analysis

We aim to provide quantitative metrics on our system's success in aligning the pre-made, score-matched animation with a realtime music performance. Because our implementation allowed the possibility of video playing backwards, evaluation was less straightforward than for a phase-vocoder-based audio accompaniment system, which usually enforces forward motion through the track. We did not find previous benchmarks to compare our performance to, but hope our findings can serve as an initial reference for future investigation of similar systems.

We performed analysis with audio from a recording of the *Wings* performance from November 29, 2022 . We obtained ground truth note onset times from the performance audio using a combination of offline score recognition and hand correction. For ease of analysis, we opted to evaluate the error of a simulated performance run instead of the video shown in the Nov 29 performance linked above. Note that in our implementation, the animation playback will sometimes differ slightly between runs – the video produced via a simulated performance was not identical to the animation seen in the live Nov 29 performance. In addition, we are running the simulation on a different device than the one used for live performances (performances were run on a Macbook Pro, the simulation was run on a Macbook Air).

To create a simulated performance, we fed the November 29 performance audio into the Informatics Philharmonic "synth mode", which performs recognition and predicts future note onset times as if the audio were being received by the system in real time. This displayed an animation that was rate-modulated in real time to align with the recorded clarinet performance. We took a screen recording of this animation, which converted the variable frame rate of video produced by the Max/MSP patch to a constant frame rate of 60 fps. We then matched the frame at each location in the screen recording to its "ground truth" counterpart in the original animation. This correspondence was created automatically by finding which ground truth frame minimized the sum of squared error for a given frame in the screen recording. Note that in Nikki's original animation, there were sometimes long sequences of frames with no movement. To aid the automatic labeling process, we used a modified animation including a large timestamp in the bottom right corner. This timestamp updated around every two frames – thus, our labeling is accurate to a resolution of 2 frames.

With labeled frames from the synthesized performance, we can now compute alignment error as the time difference between a clarinet note onset and its corresponding keyframe:

$$E_i = t_{note_i} - t_{key_i} \tag{7}$$

t_{note_i} is the onset time of the i^{th} note in the score. t_{key_i} is time when the keyframe associated with $note_i$ is displayed in the performance.

Note that even without frames playing backwards, a given keyframe may appear multiple times or not at all. For example, if the clarinetist plays twice as fast as the original animation rate, keyframes could be skipped; if the clarinetist plays half as fast as the original animation rate, keyframes could be repeated. These situations can be accounted for by slightly broadening the definition of t_{key_i}. t_{key_i} can be interpolated from existing frames if the i^{th} keyframe is missing. If there are consecutive repeated instances of the i^{th} keyframe, a single instance or the average could be chosen.

This relatively straightforward conception of alignment error is complicated by the fact that our system would sometime play animation "off course" and correct itself by playing frames backwards. For example here, the frames are well- aligned until 27 seconds into the video, where the animation starts suddenly starts playing much faster. This sudden increase in speed causes several keyframes to be played early. When subsequent prediction messages are received, the system is able to re-align the animation by quickly moving backwards and replaying the frames in a way that aligned with the clarinet performance. In the performance take used for our analysis, the backwards trajectory described above occurred several times – the video would get "off course", and the system would correct itself. A graph of the ideal, score-synchronous, keyframe locations vs. observed keyframe locations in the simulation can be seen in Figure 1c.

To some extent, this type of behavior is not unexpected because it is not explicitly prohibited by system design. However, large jumps create noticeable animation artifacts that we consider undesirable. We plan on modifying the system to explicitly prevent too-large jumps into the past or future over a short period of time.

Backwards behavior creates analysis challenges because it makes a subset of keyframes appear at multiple, disjoint locations in the displayed animation. The process of choosing the appropriate t_{key_i} is thus ambiguous.

To address this challenge, we present two methods of computing system alignment. The first method sets t_{key_i} as the time of the first frame that either matches or passes the i^{th} keyframe. Formally,

$$t_{key_i} = \min_t key_i <= f_t \tag{8}$$

Where f_t is the frame at time t. This metric penalizes course correction in instances such as the situation described above by computing error from the initially rushed trajectory before the video jumped back to better align with the clarinetist. We can see a histogram of this error metric in Figure 1a. According to this metric, 64% of keyframes appear within 0.1 seconds of their respective note onsets. Studies have found that within 0.1 seconds, humans will consider a visual stimuli synchronous to an auditory stimuli [19].

We also wish to provide a metric that does a better job capturing the perceptual alignment of the keyframes with the clarinet part. Our perceptual error metric aims to answer the question of whether a clarinet note onset *appears* aligned to its corresponding gesture in the animation. Therefore, if an instance of key_i is displayed at the same time as t_{note_i}, there should be a perceptual error of zero. Using this metric, other instances of key_i farther from t_{note_i} are disregarded.

To measure perceptual error we defined the continuous function $F(t)$ as an interpolated version of the original frame sequence S, where t is time(in seconds) where a frame in the screen recording is displayed, and $F(t)$ is the interpolated frame position. Because $F(t)$ is not a one-to-one function, there can be multiple times where $F(t) = key_i$. Thus to gauge perceptual synchrony, we choose the time that is closest to the note onset. Formally,

$$t_{key_i} = \underset{\{F(t)=key_i\}}{\operatorname{argmin}} \; |t_{note_i} - t| \qquad (9)$$

A histogram of this perceptually-based metric can be viewed in Figure 1b. Using this accuracy metric, 74% of keyframes occur within 0.1 seconds of the corresponding note onset.

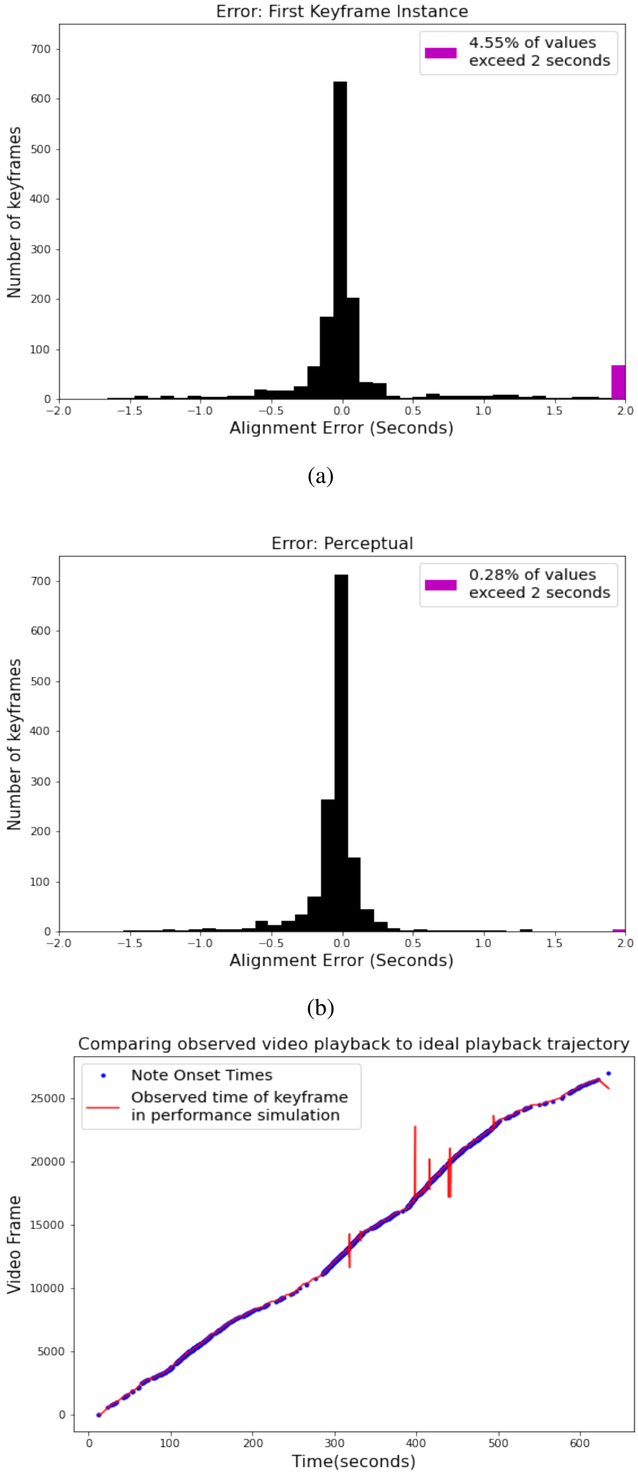
3.2 Qualitative results

In this section, we will discuss the efficacy of our system from a design level, a performance level, and share audience reception. One big takeaway we had as technology creators was that some of our seemingly minor design choices and assumptions had an immense downstream effects on how the system worked “in the field”. In Section 4, we will propose methods for mitigating observed negative effects.

3.3 Design and Performance

We created the design pipeline with two main goals in mind: 1) we wanted the ability to adapt an arbitrary musical score with little-to-no piece-specific pre-processing, and 2) we wanted to give a video producer the ability to arbitrarily choose which notes in the score are aligned with animated gestures without altering the **meaure-to-keyframe dictionary**. To a large extent, these goals were met. Nikki designed animations independently and was able to tweak them at will between performances by switching out the video file controlled by the Max/MSP patch. We were also able to use largely the same system to run *Wings* and *Fanfare for the Uncommon Woman No. 5*, with the exception of slight bug fixes for the latter to accommodate a wider range of score conventions. For example, we added support for pickup notes and mixed meter in order to correctly decode the *Fanfare* MIDI score.

However, there were instances where the ease of our pipeline meant artistic flexibility was sacrificed. One such example was the choice to have a one-to-one mapping between notes and keyframes. Certain animation effects that Nikki wanted to include were incompatible with this framework. For example, Nikki originally designed the beginning of the *Fanfare for the Uncommon Woman No. 5* animation to



(a)

(b)

(c)

Figure 1: (a) Histogram showing alignment error when computed with the first instance of the observed keyframe (b) Histogram of alignment error using perceptual error metric. For all histograms, positive error indicates a keyframe occurs early (before the note onset), while negative error indicates a keyframe occurs late (after the note onset). (c) Trajectory comparison between ideal score-synchronous keyframe display times and times keyframes were actually displayed during a simulated run.

slowly fade in, culminating at the *first* clarinet onset. However, since the beginning of the fade-in had no corresponding note in the clarinet part, it could not be specified as part of a solo-unison accompaniment. We further cautioned Nikki against adding fade-in effects or any other type of easing transition before "cues", or notes requiring the responsive pipeline described in Section 2.2. Cues are manually notated by the performer in cases where they plan to take an indeterminate-length pause. These fade-in effects violate our "one-way communication" assumption that the video will respond to the musician but the musician will not act based on the video. In order to align a fade-in before a cue, the musician would need to look at the video in order to time their entrance appropriately.

We also observed violations of the "one-way" communication assumption during the rehearsal and performance process. During rehearsal, Nikki would adjust her clarinet playing at certain instances to "optimize" the visual effect. Because Nikki wanted to create the best possible audiovisual performance, she tried to act in a way that would extract optimal behavior from the technology. During performance, Nikki and the other players usually did not respond directly to the video's behavior, instead turning their gaze to the sheet music, the audience, or other musicians. However, there were times where Nikki *did* look back at the video to confirm the performance was still in sync. For performances that occurred before the technology was stable, this allowed Nikki to know if the system was no longer aligning properly; she could then decide whether to stop and start over. However, during a later performance, Nikki was "thrown" by an instance when she looked back and the video was not exactly as she expected. This caused her to stop performing and start over. When looking at the concert recording after-the-fact, we discovered the video actually *was* perceptually in-sync with her playing at the moment where she stopped. Thus if Nikki had not looked back, the performance could have been completed without technical issues.

This example illustrates trust as an essential feature in a one-way system: an individual must trust that a system will perform as expected if they are not receiving feedback on the system's efficacy. We are lucky to have a collaborator like Nikki who is willing to perform with emerging technologies. After seeing performances of live accompaniment works, a musician in the audience reached out to Nikki to commission a video accompaniment animation. However, he preferred a "fixed media" version of the video accompaniment, where the live soloist is responsible for aligning with a static video. If a musician is not confident a technology will work during concert-time or has less technological support, they will often opt for less sophisticated technology with less potential for failure.

3.4 Audience Responses

In general, audiences appreciated the artistic value of the alignment and felt that animation enhanced their experience, with the exception of a few audience members who simply did not like the pairing of music with animation. One noted that the animation restricted his understanding of the piece to a single set of imagery. This stopped him from having the freedom to picture different mental images during the performance. Audience members expressing this type of sentiment tended to be experienced concertgoers, or worked in music-related fields. Nikki acknowledged these comments, but feels that the experience of an animation-augmented performance is fundamentally different from a clarinet solo performance – the animation gives an explicit window into *her* artistic interpretation. Because she has the freedom to adjust animation at will, she can alter the animation should she want to convey something different. Additionally, a major reason Nikki started creating animated accompaniments was to increase a performance's accessibility to non-musician audiences. Therefore, she does not see professional musicians as the direct audience being targeted by this art form.

From a technical perspective, many audience members were able to distinguish instances where the alignment was successful vs. unsuccessful.We observed this qualitatively from comparisons made by people who saw both the initial concerts where the technology "failed" mid-performance, and future performances where alignment was successful throughout. Notably, many audience members still gave positive feedback for the initial "failed" performance despite the lack of alignment, viewing even the unaligned animation as a positive addition to the music. However, audience members who attended both "failed" and "successful" performances felt the well-aligned animation had a greater impact on enhancing their performance experience. Even very young audience members were able to make the distinction between well and poorly-aligned animation. During our elementary school performance, the animation and clarinet performance was initially well-aligned, but became uncoupled halfway through. One student asked if "the robot got lost" during the performance, indicating an appreciation of the intended synchrony.

Even when the video accompaniment technology was working as expected, certain audience members noticed alignment discontinuities or temporary asynchrony around cues. In *Wings*, Nikki pre-planned positions in the score to take long breaths, marking the next note entrance with a cue indicating the system would wait for that note to be detected by the score follower before resuming video progression. One musician in the audience of the November 29 performance commented:

> *"When I was watching Wings, there were these strange moments that . . . Nikki would reach this juncture in the music where [she] had to breathe. And what happened after, the breath ... occasionally it wouldn't line up quite visually with the music . . . Is there a struggle for the technology to sense and react to your breathing?"*

This type of situation can be observed at several times in the November 29 performance recording here – for example, when Nikki takes a breath at around time 33:03 then plays her next note at around 33:04, the corresponding keyframe is not displayed until around 33:05.

Misalignment at cues can arise for two reasons. First, the animation could reach the cue's keyframe before Nikki plays the cued note, causing video playback to stop and wait for her entrance. Additionally, the reactive alignment strategy used at cues involves latency. Depending on factors like the starting volume of the cued note, Informatics Philharmonic may need an audible amount of time after the cued note onset occurs to determine the note has been played and transmit that information to the Max/MSP patch. This in turn causes the video to react late.

The above comment also gives insight into the way this audience member assigned meaning to his observations of the AI system's behavior. His proposed explanation for the misalignment was that the system struggled to detect breaths. This is likely because quick breaths are commonly used by clarinetists to signal they will begin playing. Since our score follower is only based on pitches notated in the score, this "breath" communication obvious to a human collaborator is ignored by our system.

We highlight this example because in the absence of detailed technical knowledge of a system, users will form "mental models" to explain how a system works and base their behavior off that mental model [20]. If a musician using the technology assumed that a "breath detection" system sometimes "struggled", they could take unnecessary mitigating action such as taking louder, more obvious breaths. It is important that we convey enough information clearly to users so they do not build faulty mental models.

Conversely, a mental model that aligns with the reality of the system can allow for positive artistic outcomes. Nikki communicated with us throughout her animation and performance process, and thus had an accurate mental model of cueing. This knowledge affected the way that she designed and performed *Wings*. Nikki tried to create animation in a way that masked potential discontinuities in the animation, for example not including moving figures right before cues. Nikki also honed her interpretation of the music in a way that would make the animation "look good". For example during slow section where every note was cued, she tried to time here entrances so the delay seemed natural.

4. CONCLUSIONS AND FUTURE WORK

When designing this system, we made an assumption that the video "accompaniment" was completely subservient to the human performer, thus only one-way communication from human to the accompaniment system was necessary. Our experiences working with video accompaniment showed that this was not the case in three types of situations:

1. *Inappropriate specification:* The video may need to have "keyframes" outside of score-synchronous times, e.g. to start a "fade-in" effect before a cue.

2. *Masking issues in technology performance:* The performer alters their musical interpretation in order to mask discontinuities caused by cueing.

3. *Lack of Trust:* The performer checks back at the video during performance to check if the system is working properly.

We will discuss how these three situations can be addressed to improve the video accompaniment system.

4.1 Alternatives to One-Way Specification

One could trivially add extra keyframes to signal the start of transitions such as "fade-ins" with an external communication device like a foot pedal – the musician would hit the pedal once to trigger the fade-in, then switch back to note onset-based control. This solution maintains the concept of one-way control, but may not be appropriate for artistic or practical reasons. Requiring consistent operation of an additional piece of hardware during a performance adds additional tasks for the musician to complete and increase the chance of technological errors during performance. A better solution could be utilizing two-way communication during these instances. Before a cued note, the musician would look back at the video to gauge when their next onset should be. Using this framing, the musician would respond to the video at cues preceded by easing transitions, and the video would respond to the musician in all other circumstances. This type of "cue switching" control has been previously implemented in automated accompaniment systems such as MuEns – in certain situations, musicians time their cue entrance in response to movement of a projected visualization; in other situations, the accompaniment system times its cued entrance in response to the live musicians [21].

In certain artistic situations, one may also want enforce that the video maintains a constant frame rate across multiple note onsets. For example, imagine again the ball-tossing video described in Section 2, but this time the corresponding solo part is a quick sequence of notes. Unless the musician plays all the notes in the sequence with exactly the same length, the ball's acceleration would become jerky and unnatural. We can observe this issue in certain parts of the November 29 Wing's performance. One can see here how a series of swooping wings corresponding with clarinet runs sometimes have choppy rather than smooth motion. Depending on the artistic use case, it may make sense for either the musician follow the video, or make the video accompaniment system align to only a subset of solo note onsets (i.e. only have keyframes corresponding to the first and last note of a run).

4.2 Improving alignment and smoothness

We have observed that when the video accompaniment is not as aligned or smooth as the performer wants, they will alter their musical interpretation in order to present the best combined audiovisual performance. We consider this type of interaction undesirable, as it involves the musician compromising their interpretation to accommodate the technology. Improving the system's performance would mitigate the need for this this type of adjustment. Here, we will mainly discuss how to improve performance at cues.

Aside from modifying our score follower to parse "cue signals" such as sharp breath intakes, we see two ways to improve the video experience around cues. The first approach is creating a more fluid way to deal with video playback at cues instead of just pausing the video until the cue is reached. As described in Section 2.1, the vanilla Informatics Philharmonic slows audio down in an exponential fashion so frames continue advancing before a cue. We could implement a similar system for video so motion never stops, thus making cues potentially less jarring and "softening" the visual effect of a slightly late cue detection. Another potential direction is decreasing the response time of the reactive pipeline. For a faster reaction speed, we could lower the amount of certainty the system needs to determine a note has been played, thus leading to a faster reaction time. However, this approach has the potential of increasing the chance of a "false positive" detection if the performance environment is noisy. Currently, our system is able to still perform well in noisy environments: one of our successful *Wings* concerts was held outdoors on a windy day. We hope to explore ways of increasing reaction time without sacrificing robustness.

4.3 Trustworthy system

We also want to add features to make our system more "trustworthy" for performers. We plan on incorporating in a higher-level communication and control system that a live performer can use during concert time to control the flow of the performance. This way, the performer can 1) receive confirmation that the technology is working, and 2) take mitigating action if they perceive a breakdown in alignment. A confirmation system can be something as simple as the system's perception of the current measure number being displayed to the soloist. This way, the soloist can quickly confirm the system's position matches their position in the music without worrying about alignment details. We can also implement a foot pedal system similar to many existing contemporary electronic pieces, where the performer can use the foot pedal to "jump" to a different section of the media. For example, Russell Pinkerton's flute and electronics composition *Lizmander* has sections where electronic accompaniment effects are triggered by the detection of certain flute pitches. However, should the automatic triggering fail, the performer is encouraged to "manually advance the program by pressing the foot switch" [22]. This type of easy-to-use failsafe mechanism can give musicians confidence that potential technological issues will not derail their performance. Of course there is an inherent "leap of faith" required to work with emerging technologies during concert, but we hope to create systems that actively inspire confidence and make musicians more willing to welcome new performance technologies.

Acknowledgments

We thank the audience members from our performances for their support and feedback.

5. REFERENCES

[1] Nikki Pet. "Moro, lasso" - Carlo Gesualdo — Nikki Pet, clarinet. Youtube. [Online]. Available: https://www.youtube.com/watch?v=5uO7-1EMrTM&ab_channel=claripet

[2] ——. Hommage à R. Strauss, Béla Kovács — Nikki Pet, clarinet. Youtube. [Online]. Available: https://www.youtube.com/watch?v=qQg2T0DXBF8&ab_channel=claripet

[3] B. Vercoe, "The synthetic performer in the context of live performance," in *Proceedings of the International Conference on Computer Music (ICMC)*, 1984, pp. 199–200.

[4] R. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proceedings of the International Conference on Computer Music (ICMC)*, 1984, pp. 193–198.

[5] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music." in *International Computer Music Conference (ICMC)*, Belfast, Ireland, Aug. 2008, pp. 33–40. [Online]. Available: https://hal.inria.fr/hal-00694803

[6] P. Cuvillier and A. Cont, "Coherent time modeling of semi-markov models with application to real-time audio-to-score alignment," in *IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2014, Reims, France, September 21-24, 2014*. IEEE, 2014, pp. 1–6. [Online]. Available: https://doi.org/10.1109/MLSP.2014.6958908

[7] 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," vol. 8610, 08 2014, pp. 134–145.

[8] A. Arzt, G. Widmer, and S. Dixon, "Automatic page turning for musicians via real-time machine listening," 01 2008, pp. 241–245.

[9] E. Gomez, M. Grachten, A. Hanjalic, J. Janer, S. Jorda, C. F. Julia, C. Liem, A. Martorell, M. Schedl, and G. Widmer, "PHENICX: Performances as Highly Enriched aNd Interactive Concert Experiences."

[10] C. Raphael, "Music Plus One and Machine Learning," in *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, 2010, pp. 21–28.

[11] A. Cont, P. Cuvillier, J. Echeveste, and L. Tan Van Liu, "Metronaut." [Online]. Available: https://www.metronautapp.com/

[12] Cycling '74, "Max/MSP," 2023. [Online]. Available: https://cycling74.com/products/max

[13] *The Informatics Philharmonic in New Music*. Zenodo, Jun. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5043214

[14] M. Prockup, D. Grunberg, A. Hrybyk, and Y. E. Kim, "Orchestral performance companion: Using real-time audio to score alignment," *IEEE MultiMedia*, vol. 20, no. 2, pp. 52–60, 2013.

[15] A. Cont, P. Cuvillier, J. Echeveste, and J.-L. Giavitto, “Antescofo.” [Online]. Available: https://forum.ircam.fr/projects/detail/antescofo/

[16] A. Cont, J. Echeveste, J.-L. Giavitto, and F. Jacquemard, “Correct automatic accompaniment despite machine listening or human errors in antescofo,” in *International Conference on Mathematics and Computing*, 09 2012.

[17] G. Xia, M. Kawai, K. Matsuki, M. Fu, S. Cosentino, G. Trovato, R. Dannenberg, S. Sessa, and A. Takanishi, “Expressive humanoid robot for automatic accompaniment,” in *SMC 2016 - 13th Sound and Music Computing Conference, Proceedings*, ser. SMC 2016 - 13th Sound and Music Computing Conference, Proceedings, R. Grossmann and G. Hajdu, Eds. Zentrum fur Mikrotonale Musik und Multimediale Komposition (ZM4), Hochschule fur Musik und Theater, 2019, pp. 506–511.

[18] M. S. Cuthbert and C. Ariza, “Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data.” in *Proceedings of the 11th International Society for Music Information Retrieval Conference*. Utrecht, Netherlands: ISMIR, Aug. 2010, pp. 637–642. [Online]. Available: https://doi.org/10.5281/zenodo.1416114

[19] V. Arstila, A. L. Georgescu, H. Pesonen, D. Lunn, V. Noreika, and C. M. Falter-Wagner, “Event timing in human vision: Modulating factors and independent functions,” *PLoS ONE*, vol. 15, 2020.

[20] P. Johnson-Laird, “Mental models in cognitive science,” *Cognitive Science*, vol. 4, pp. 71 – 115, 1980.

[21] A. Maezawa and K. Yamamoto, “MuEns: A Multimodal Human-Machine Music Ensemble for Live Concert Performance,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Denver Colorado USA: ACM, May 2017, pp. 4290–4301. [Online]. Available: https://dl.acm.org/doi/10.1145/3025453.3025505

[22] R. Pinkston, “Lizmander,” 2003. [Online]. Available: http://www.russellpinkston.com/?portfolio=item-three