Expressive Collaborative Music Performance via Machine Learning

Gus (Guangyu) Xia

CMU-ML-16-103

August 2016

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Roger Dannenberg, Chair
Geoff Gordon
Larry Wasserman
Arshia Cont

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under grant number IIS0855958 and by a gift from Google.

Keywords: Expressive performance, human-computer interaction, automatic accompaniment, spectral learning, humanoid robot

To my parents and my girlfriend.

Abstract

Techniques of Artificial Intelligence and Human-Computer Interaction have empowered computer music systems with the ability to perform with humans via a wide spectrum of applications. However, musical interaction between humans and machines is still far less musical than the interaction between humans since most systems lack any representation or capability of *musical expression*. This thesis contributes various techniques, especially machine-learning algorithms, to create artificial musicians that perform *expressively* and *collaboratively* with humans. The current system focuses on three aspects of expression in human-computer collaborative performance: 1) expressive timing and dynamics, 2) basic improvisation techniques, and 3) facial and body gestures.

Timing and dynamics are the two most fundamental aspects of musical expression and also the main focus of this thesis. We model the expression of different musicians as co-evolving time series. Based on this representation, we develop a set of algorithms, including a sophisticated *spectral learning* method, to discover regularities of expressive musical interaction from rehearsals. Given a learned model, an artificial performer generates its own musical expression by interacting with a human performer given a pre-defined score. The results show that, with a small number of rehearsals, we can successfully apply machine learning to generate more expressive and human-like collaborative performance than the baseline automatic accompaniment algorithm. This is the first application of spectral learning in the field of music.

Besides expressive timing and dynamics, we consider some basic improvisation techniques where musicians have the freedom to interpret pitches and rhythms. We developed a model that trains a different set of parameters for each individual measure and focus on the prediction of the number of chords and the number of notes per chord. Given the model prediction, an improvised score is decoded using nearest-neighbor search, which selects the training example whose parameters are closest to the estimation. Our result shows that our model generates more musical, interactive, and natural collaborative improvisation than a reasonable baseline based on mean estimation.

Although not conventionally considered to be “music,” body and facial movements are also important aspects of musical expression. We study body and facial expressions using a humanoid saxophonist robot. We contribute the first algorithm to enable a robot to perform an accompaniment for a musician and react to human performance with gestural and facial expression. The current system uses rule-based performance-motion mapping and separates robot motions into three groups: finger motions, body movements, and eyebrow movements. We also conduct the first subjective evaluation of the joint effect of automatic accompaniment and robot expression. Our result shows robot embodiment and expression enable more musical, interactive, and engaging human-computer collaborative performance.

Acknowledgments

The story begins in March 2009, when Bryan (Minghui) Bi, my music teacher at Peking University, first introduced me the concept of "Computer Music". He urged me to check out Roger Dannenberg's work and to do an internship at Wen Gao and Xihong Wu's lab. I was soon captivated by this field during the internship, and later on extremely honored to be a Ph.D. student at Carnegie Mellon University. I am grateful for every moment in the past seven years: day and night, happiness and sorrow, success and failure.

First and foremost, I am very grateful to Roger's fatherly advice. He teaches me when I am naive, supports me when I am weak, and encourages me to find my own direction when I can stand on my own. The closer I approach graduation, the more I realize that what I have learned from him is not only the way to perform research but also a philosophy of life — to combine personal interests with human kindness and carry it on in a lifelong career. I am also beholden to Geoff Gordon, Larry Wasserman, and Arshia Cont for serving on my thesis committee. It would be impossible to complete this thesis without the machine learning methods I learned from Geoff, the statistical tools I learned from Larry, and the cutting-edge automatic accompaniment systems developed by Arshia.

During the six-year Ph.D. study, I am particularly fortunate to receive extra advice from many other world-class scientists and artists, such as Raj Reddy, Tom Mitchell, Manuela Veloso, Hui Zhang, Bhiksha Raj, Christos Faloutsos, Avrim Blum, Rich Stern, Eric Xing, Daniel Sleator, Stephen Neely, Riccardo Schulz, Michael Casey, Byron Boots, Salvatore Sessa, Atsuo Takanishi, Michael Friedmann, Bruno Repp, and Jinhua Zhao. From the conversations with them, I saw a more complete picture of Computer Science and its relationship with other fields.

At Carnegie Mellon, I am also fortunate to share my life as a graduate student with many talented young researchers and musicians, such as Dawen Liang (大师兄), Yun Wang (天才), Yuan Zhou (大师), Xingguo Li, Abulhair Saparov, Calvin McCarter, Anders Øland, Boyan Li, Zeyu Jin, Jiuqiang Tang, Haochuan Liu, Mutian Fu, Tongbo Huang, Yifei Ma, Laxman Dhulipala, Yuxiang Wang, Yaoliang Yu, Fei Liu, Melody Fu, Yiqian Song, Tina Liu, Andy Wang, Peng Peng, Xizi Xu, Zhipeng Yang, Michael Moody, Junchen Jiang, Xuezhi Wang, Ruogu Kang, Xi Chen, Xi Liu, Nan Li, Lei Li, Yanlin Li, Yuandong Tian, Yajie Miao, Hao Zhang, Guangyao Miao, Duo Ding, Bin Liu, Mingche Su, Zhubin Huang, Shang Wang, Hanzhi Chen, and Xuege Zhang. My Ph.D. journey would be incomplete without these friendships. Especially, I own many thanks to Shan Qi, Hannah Criswell, and Aakansha Vishnoi for their care, tolerance, and encouragement.

Last but not least, I would like to thank Diane Stidle, Sharon Cavlovich, and Christina Contreras for their assistance and would like to thank Wanli Fang (领导) for her tremendous support, unyielding patience, and the constructive challenges she has been designing for me to overcome.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Music has served human beings for thousands of years as a universal language and unique medium for expression. For humans, music means more than sound. “Music expresses that which cannot be said and on which it is impossible to be silent [1].” Under the surface of the acoustic signal, *expression* conveyed via music subtly speaks to our moods, our emotions, and even our views on the universe and life. Just as poetry has a symbolic representation on paper as well as the possibility of vocal realization and interpretation, music exists in symbolic notation form (the score), yet relies on interpretation by musicians through *expressive performance*.

An expressive performance deviates from the mechanical rendition of its score along various dimensions, including pitch, tempo, loudness (dynamics), articulation[1], etc. Although not conventionally considered to be “music,” body and facial movements are also important expressive elements that affect the perception and appreciation of the audience. Importantly, such deviations are not random; they follow certain patterns. Performers use these patterns of sound and motions as a powerful tool to convey and shape their musical expression. As written in the preface of *Shijing* (or *Classic of Poetry*), an ancient collection of Chinese poetry[2], “to expressive yourself: where words fail, intonation speaks; where intonation fails, music speaks; where music fails, gesture speaks.”

In most cases, a musician does not perform on stage alone. Musicians form an ensemble (or a band) to perform as a group, in which they coordinate their musical expressions to achieve a shared performance. We call this a *collaborative music performance*. In a collaborative performance, the art for the musicians is not only to perform expressively on their own, but also to keep in concert with each other as a coordinated and organic whole by continuously adjusting their musical expression. In

[1] A possible combination of note duration, loudness, and a certain performance technique (such as plucking string) that affects the sound quality of a performed note.
[2] My translation of *Shijing* is inspired by the quote “where words fail, music speaks.” by H.C. Anderson.

other words, musicians have to achieve two seemingly conflicting goals. For example, expressive timing deviations by each member of the ensemble are constrained by the overall necessity of ensemble synchronization. To sense and interact with each other's musical expression is not only crucial but also difficult. In practice, it is almost impossible to achieve satisfactory collaboration on the first performance.

To solve this problem, musicians *rehearse.* Musicians spend time practicing together by going through the pieces multiple times. Through rehearsals, musicians become familiar with the style of each other while setting the "communication protocols" for musical expression. For example, when should one musician play with more tempo fluctuations, and when should another musician keep a steady beat? What is the desired trend and balance of dynamics? It is important to note that these protocols are usually complex and implicit in the sense that they are difficult to express via explicit rules. (Musicians in a large ensemble even need a conductor to help set the protocols.) However, musicians are able to learn these protocols very effectively. After a few rehearsals, professional musicians are prepared to handle new situations that do not even occur in rehearsals, which indicates that this learning procedure goes beyond mere memorization.

Although deeply explored experientially by musicians, the complex mechanism of sensing and coordinating music expression in collaborative performance has not been well understood by science yet. We still know little about the structure or computation behind expressive performance and the cognitive strategies that enable musicians to learn so effectively and efficiently from rehearsals. As both a computer scientist and a professional musician, I believe that the best way to understand expressive collaborative performance is to create it, i.e., *to create artificial performers that are able to perform expressively in concert with human musicians*. From a scientific perspective, advancement in artificial performance and artificial performers can help us to better understand *what is musical expression* and *how humans process music*. From a humanistic perspective, this work can even enlighten us as to *what music means to humans* and *who we are*. From an application perspective, collaborative artificial performers can serve both amateur and professional musicians for music practice, on-stage performance, and even music tutoring.

In particular, the procedure of learning musical interaction through rehearsal suggests that a computer system can be trained in a similar way using computational techniques, especially machine-learning algorithms. Hence, this thesis is titled *Expressive*

Collaborative Music Performance via Machine Learning. The remainder of this chapter is organized as follows:

- Section 1.1 provides a broad background for this thesis topic, where we motivate the thesis study and highlight its significances from the perspectives of Artificial Intelligence, Human-Computer Interaction, and Music Performance.
- Section 1.2 narrows the view back into computer-aided music performance, where we review the current state of research and highlight the connections and differences between this thesis and previous related work.
- Section 1.3 presents the thesis statement, where we formally define the problem from a machine-learning point of view and list the specific research questions we will answer in this thesis.
- Section 1.4 gives an overview of the rest of the thesis, where we present the main methodology and contribution of each chapter.

1.1 Background and motivation

The study of expressive and collaborative music performance via machine learning is intrinsically interdisciplinary. It lies within the intersection of *Artificial Intelligence* (AI), *Human-Computer Interaction* (HCI), and *Music*. Figure 1 illustrates such overlaps, where we see the thesis study lies in the white intersection. Some related interdisciplinary fields include: *social robotics* that lies in AI ∩ HCI, *music information retrieval* (MIR) that lies in AI ∩ Music, and *new interfaces for musical expression* (NIME) that lies in HCI ∩ Music.

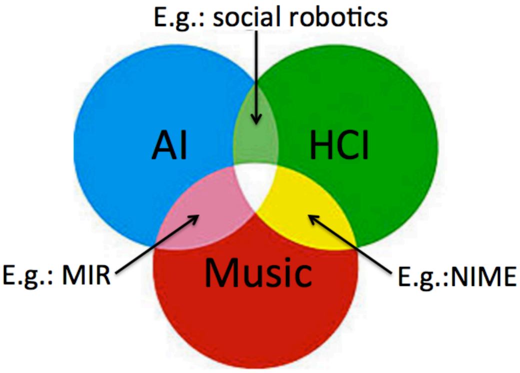


Figure 1. An illustration of the connections between this thesis and AI, HCI, and Music.

Importantly, the interdisciplinary nature of this thesis points to possible future directions of AI, HCI, and Music:

The impact on Artificial Intelligence: Intelligent systems have already reached the level of human capabilities for many tasks, including chess and board games, driving aircraft and vehicles, and even speech and face recognition. However, these tasks are almost exclusively functional; for the tasks requiring an understanding and expression of inner human feeling, such as natural language and music, computers are still far behind. To some extent, the expression and semantics of music are even subtler and more subjective than natural language. The study of musical expression, one of the most profound manifestations of humanity, is at the frontier of artificial intelligence and can potentially push the future of intelligent systems to a more aesthetic and artistic level.

The impact on Human-Computer Interaction: Computer systems that can communicate with humans through musical expression provide intimate and natural human-computer interactions. Such systems no longer need to explicitly define commands or to design extra interfaces like keyboards and touchscreens, since we humans are interacting with machines as we are interacting with each other. In other words, the boundary between humans and machines is further blurred; the expressive collaborative performance will be one of the ultimate forms of human-computer interaction.

The impact on Music: Throughout history, technology has dramatically influenced music performance. Ancient techniques of drilling small holes in bones brought humans the first flute, metallurgy and machining enabled the woodwind and brass instruments found in the modern orchestra, and electronic technology boosted the power of electric guitars and led to new forms of pop music. While intelligent systems are increasingly serving music listening, e.g. music recommendation systems, they still struggle to serve music performance since it is difficult to model and react to human musical expression in real time. On the other hand, once artificial performers are equipped with the ability to sense and interact with musical expression, they can step into people's daily lives, serving us through off-stage rehearsals, on-stage performances, and even help teach music to humans in the future.

1.2 The current state of computer-aided music performance

Computer music systems have achieved a wide spectrum of application in music performance, ranging from fixed media to free improvisation. The broad range of practice includes Music Minus One (fixed media such as karaoke), score following and automatic accompaniment (for western classical music), human-computer music performance (for pop music), and interactive computer music (for experimental music). These systems have been able to interact with humans at different aspects of performance, such as scores, phrases, notes, beats, and gestures.

However, the musical interaction between humans and machines is still far less musical than the interaction between human musicians, since most systems lack representations and capabilities of musical expression. An example can be seen from the development of score following and automatic accompaniment systems, which aim to track and accompany human soloists in real time based on a pre-defined score. Though the systems have existed for decades now, many researchers still adopt the original synchronization-oriented accompaniment model or merely refer to the problem as "score following," as if all timing and performance information derives from the (human) soloist and there is no performance problem. In a professional setting, even the term "automatic accompaniment" diminishes the complex collaborative role of performers playing together by suggesting that the (human) soloist is primary and the (computer) accompanist is secondary. Music schools usually refer to human piano accompaniment as "collaborative piano" to highlight the accompanist's importance in shaping musical expression through joint performance. In order to successfully create a collaborative music performance, all performers are equal with respect to musical expression, including the artificial performers.

There is a large gap between music practice and computer music research on the topic of collaborative music performance. This thesis aims to empower computer music systems with *music intelligence* to master expressive musical interactions. In particular, the goal is to incorporate music intelligence into the existing framework of automatic accompaniment systems, extending the system's ability from passive sight-reading synchronization to actively mimicking the expressive behavior of ensemble musicians. As a pioneering study, this thesis does not consider *all* aspects of musical expression. We consider the aspects of expressive timing & dynamics deviations, basic improvisation techniques, and facial & body gestures.

1.3 Thesis statement

Thesis statement: *With computational techniques, especially advanced machine learning algorithms, we can create an intelligent artificial performer that is able to understand and respond to human musical expression*.

There are many issues to be addressed. Questions regarding the *data and feature representations* include:

- How should we extract proper features to represent musical expression of different scales of music structure? E.g., how to represent a *crescendo* (becoming louder) phrase and how to represent a *ritardando* (slowing down) measure?
- How should we choose the level of abstraction to generate artificial performance? E.g., should we decode the artificial performance note-by-note, chord-by-chord, or measure-by-measure?
- What are the dominant aspects of music performance that affect the expressive interaction? E.g., is expressive timing affected more by rhythm or by pitch contour?

Questions regarding the *learning task* include:

- How can we design machine-learning algorithms to distill models from rehearsals? In other words, how can we learn regularity from seemingly irregular data?
- What are the limits of validity of the learned models? E.g., which model generalizes across a whole piece or even across different pieces of music, and which model only applies to some specific score locations?
- How does performance/performers' style affect the learning? E.g., would the models trained on similar performances or the same performers lead to better results?
- How many rehearsals are needed to train the artificial performer, and how does the number of rehearsals affect the learning? E.g., would the minimum number of required rehearsals be reasonable in practice, and would adding more rehearsals lead to better results?

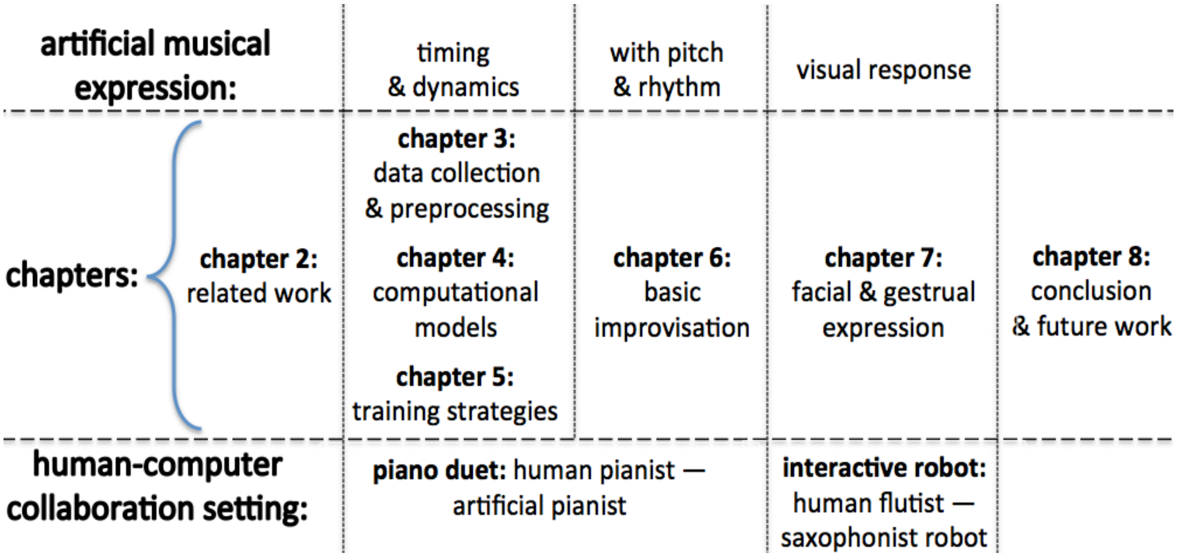
Questions regarding the *evaluation scheme* include:

- How should we design objective and subjective measurements to validate the generated artificial performance?
- How much better is the generated performance compared with our baseline, and how far/close is it from human performance?
- Is the evaluation of the artificial performance consistent under different measurements and criteria?

1.4 Thesis overview

To address to questions above, this thesis uses machine-learning algorithms to create expressive timing & dynamics deviations and basic improvisation techniques for an artificial pianist in human-computer piano duets. In addition, the thesis uses rule-based algorithms to create gestural and facial expressions for a humanoid saxophonist robot that performs with a human flutist.

Figure 2 together with the following list show the organization and an overview of the rest of the thesis:



artificial musical expression:		timing & dynamics	with pitch & rhythm	visual response	
chapters:	**chapter 2:** related work	**chapter 3:** data collection & preprocessing **chapter 4:** computational models **chapter 5:** training strategies	**chapter 6:** basic improvisation	**chapter 7:** facial & gestrual expression	**chapter 8:** conclusion & future work
human-computer collaboration setting:		**piano duet:** human pianist — artificial pianist		**interactive robot:** human flutist — saxophonist robot	

Figure 2. The organization of the rest of the thesis.

- Chapter 2 reviews related literatures in three research fields: Computer Music, Machine Learning & Statistics, and Music Psychology. Through the exiting achievements, we show that these three fields of related work are pointing in the direction of this thesis as a joint force.

- Chapter 3 describes the data collection and preprocessing techniques. We contribute a piano duet performance dataset, which is the first interactive performance dataset that contains multiple performances for each piece of music. We also contribute a preprocessing method that automatically detects and corrects performance errors, and show that the collected dataset has low error rate compared with other performance datasets.

- Chapter 4 presents the feature representation for piano duet performance, i.e., the intermediate layer between the data and the computational models. We contribute a general feature-extraction scheme for music performance. We also contribute a method that derives the onset and dynamics features of a chord based on a real performance where notes do not really begin simultaneously or have exactly the same dynamics.

- Chapter 5 presents the first set of computational models that learn from human piano duet rehearsals and generate artificial performances with expressive timing and dynamics given human performances. We contribute the first spectral method that learns a linear dynamic system with group lasso regularization. We also contribute different measurements to evaluate the generated artificial performers and show that the best machine-learning model, with a small number of rehearsals, consistently outperforms the baseline.

- Chapter 6 presents the different training strategies under realistic constraints of data collection. We show how the experimental results are affected by general performance rules, composition-specific structures, performer preferences, and performance styles.

- Chapter 7 describes how to learn basic improvisation techniques from piano duets. We contribute the first piano duet improvisation dataset, containing multiple improvisations for each piece of music.

- Chapter 8 presents how to generate interactive facial and gestural expressions. We contribute the first interactive music performance between a human musician and a humanoid music robot. We also contribute the first subjective evaluation on the joint effect of automatic accompaniment and robot expression.

- Chapter 9 presents the conclusion and future work.

The results presented in this thesis were developed in collaboration with various of co-authors, including: Sarah Cosentino, Roger Dannenberg, Mutian Fu, Geoffery Gordon, Mao Kawai, Kei Matsuki, Salvotore Sessa, Atsuo Takanishi, Gabriele Trovato, Yun Wang, and Larry Wasserman.

Chapter 2

Related Literature

Related work comes from three different research fields: *Computer Music*, where we find the antecedents of and inspiration for this thesis; *Machine Learning* & *Statistics*, where we find the schemes and tools to solve the defined problems; and *Music Psychology*, where we find the musicological insights that help to design better computational models.

2.1 Computer music related work

We review three realms of related work in computer music: *score following and automatic accompaniment*, *expressive performance*, and *music robotics*. The first realm focuses on collaborative performance between humans and machines, the second one focuses on artificial musical expression, and the last one focuses on physical gestures and movements. All these three realms have been developing for more than 20 years but never really informed each other; this thesis bridges these three areas.

2.1.1 Score following and automatic accompaniment

In 1984, score following and automatic accompaniment systems were independently introduced by Dannenberg [2] and Vercoe [3]. Given a pre-defined music score, these systems were able to follow a musician's monophonic performance in real time and output the accompaniment by strictly following the musician's tempo. Dannenberg's work was soon commercialized by SmartMusic and has been used by thousands of students for music practice. Ever since then, many extensions [4]-[7] have been made by Dannenberg and his collaborators. Bloch and Dannenberg [5] developed fast methods for following polyphonic performance input; Grubb and Dannenberg [6] extended this idea further to handle ensemble performance input. Later on, Grubb and Dannenberg [7] developed the first stochastic method for tracking a vocal performer. More recently, several advanced probabilistic models have been introduced [8]-[11] for more robust score following.

While most attention has been given to the "score following" (performance-score matching) part of the system, the musical expression of artificial performers or the "automatic accompaniment" part has been overlooked. As a consequence, while "score following" has already become a more-or-less solved problem, recent systems still compute accompaniment timing by linear score-performance time mapping and extrapolating to the next note (which was introduced 30 years ago). In other words, computer systems are still "passive" in the sense that they do not have any particular knowledge about performance to actively predict human behaviors. In addition, computer accompaniment systems have addressed the problem of music synchronization almost exclusively, without regard for other aspects of musical expression.

Dannenberg's original work clearly stated at the beginning of the "Limitations" section that, "the present set of algorithms make no attempt to adjust tempos in a particularly musical manner… Furthermore, no effort has been made to respond to the soloist in any way other than temporally. For example, a human accompaniment is expected to respond to loudness, articulation, and other nuances in addition to temporal cues." In addition, Desain and Honing [12] discovered that expressive timing in music performance *does not* generally scale proportionally with tempo. Both studies are suggesting that the accompaniment problem needs to be considered more seriously.

As far as we know, Raphael's Music Plus One [8] and IRCAM's AnteScofo [9] are the only two systems that consider the accompaniment problem. The former trained a Bayesian network by rehearsals to achieve more precise synchronization; the latter used a synchronization model based on Large's work [10] to achieve more natural tempo adjustment. However, the perspective is still limited to temporal synchronization; the computer's active role in shaping different musical expression is not yet considered.

2.1.2 Expressive performance

The discipline of expressive performance studies how to convert static scores into human-like expressive performances by different computational models (See Kirke's survey [13] and Widmer's review [14] for more comprehensive overviews.) The models fall into three main categories, which are rule-based modeling, case-based modeling, and probabilistic modeling. Generally speaking, probabilistic modeling works better than the others, and there is evidence that even better performance can be achieved by combining different models.

Rule-based systems, appearing in the early 1980s, generate performances based on defined or discovered performance rules [15]-[22]. Sundberg and his collaborators built the well-known KTH model by an innovative "analysis-by-synthesis" approach in which musicians and researchers worked together [15]-[18]. Researchers created models to "perform" music, which was then critiqued by expert musicians, leading to a cycle of refinement and evaluation. Others discover rules by collecting measurements from actual performance data. Among them, Todd [19][20] focused on the relationship between music structure and performance. Widmer developed various data mining methods [21][22] to discover rules from data automatically. Since the late 1990s, cased-based reasoning systems have appeared, which generate performances by adopting previous performance examples. Two representative ones are the SaxEx system [23] developed by Arcos et al., and the DISTALL system [24] developed by Widmer and Tobudic.

More recently, we see probabilistic modeling systems [25]-[28]. Generally speaking, these systems model the conditional probabilistic distribution of the performance given the score, and then generate new performances by sampling from the learned models. Grindlay and Helmbold [26] used *hidden Markov models* (HMMs) and learned the parameters by a modified version of the Expectation-Maximization algorithm. Kim et al. [27] used a *conditional random field* (CRF) and learned the parameters by stochastic gradient descent. Most recently, Flossmann et al. [28] used a very straightforward linear Gaussian model to generate the musical expression of every note independently, and then used a modification of the Viterbi algorithm to achieve a smoother global performance.

All these probabilistic modeling systems successfully incorporate musical expression with computational models. From the machine learning perspective, the underlying graphical models used in these studies serve as a good basis for this thesis. Notice that our work considers not only the relationship between score and performance but also the interaction between different performers. From an optimization point of view, the studies on expressive performance aim to optimize a performance given a score, while this thesis aims to solve this optimization problem under the constraints created by the performance of other musicians. Also, we are dealing with a real-time scenario that does not allow backward smoothing or any kind of post-processing.

Another interesting problem of expressive piano performance is *rolled* (or *arpeggiated*) chords. There are fewer studies related to this problem. From an analysis perspective, Repp [29] investigated some descriptive properties of arpeggiated chord onsets by using a single piece of music. To be more specific, this study considers the relative onset timing and the inter-onset-intervals within arpeggiated chords. Repp compared the results between the performances by students and experts and concluded that arpeggiating patterns are subject to large individual differences. From the synthesis perspective, Kim et al. [27] predicted the onsets of a rolled chord by first estimating the onset of the highest/lowest note and then adding intervals for the onsets of the other notes in the chord.

While both studies emphasized the behavior of each individual note within a chord, we emphasized the ensemble effect of the chords, i.e., how the notes in a chord behave as an organic whole. From an analysis perspective, we conducted subjective measurements to figure out what single onset time and dynamic level can functionally and perceptually replace the onsets and dynamics of the performed notes in a chord. From a synthesis perspective, we focused on the generation of that single onset/dynamics (rather than generating each note) and hence reduced the burden of learning parameters for individual notes.

2.1.3 Music robotics

Related work in music robotics can be categorized according to two perspectives: non-humanoid vs. humanoid, and pre-programmed vs. interactive. Our study belongs to the interactive humanoid robot category.

Non-humanoid vs. humanoid: Musical player robots play an important role in the study of musical interactions. Non-humanoid music player robots with interaction capabilities, such as Shimon [30] and Haile [31], have been used extensively to test fundamental musical interaction models. We believe that non-verbal human gestures can be mimicked exquisitely and replicated by robots to better study the influence of embodiment in musical interaction. Therefore, we used a humanoid robot in this thesis as a tool to subjectively measure musical interaction with gestural and facial expression.

Pre-programmed vs. interactive: While most music robots have the potential to adapt their performance to others, most of their performances are still pre-programmed. We started to see more interactive music robots developed in the past decade. Generally, these robots detect beats from music and adjust their behaviors to stay synchronized with the music. These systems include an interactive dancer [32], theremin player [33], singer [34], drummer [35], marimba player [30], and other percussion players [36]. However, very few of them react to music with gestural expression or have been evaluated experimentally by human subjects. As far as we know, the only subjective evaluation for interactive music robots was done in the work on Shimon [30]. This work showed that the visual contact with the marimba robot improves audiences' subjective ratings. On top of this, our study incorporates humanoid gestural and facial expression and conducts the first evaluation to inspect the joint effect of robot expression and music interaction.

2.2 Machine learning and statistics related work

To learn musical expression in collaborative performance, we used regression and time-series modeling as the main machine learning algorithms. Regression methods were used to learn the expression of static and individual notes, while time-series methods were used to construct a dynamic and smooth expression trajectory. To evaluate the generated artificial performances, we used analysis of variance (ANOVA), in particular, the one-way within-subject ANOVA as an important statistical tool to compare the subjective ratings of different models.

2.2.1 Regression with different regularizations

We first assumed that the musical expressions of adjacent notes are correlated and used linear regressions [37][38] to model their relationship. To further treat music performance as a time-invariant system (i.e., all notes share the same model), many other features have to be extracted to represent each note. This leads to a high dimensional feature space and hence requires different regularization techniques [39]-[41]. In particular, group lasso [40][41] fits the music performance problem very well since our feature space can be decomposed into different groups, such as "score" vs. "performance" and "timing" vs. dynamics.

2.2.2 Spectral learning for linear dynamic systems

The most sophisticated learning approach explored in this thesis is a spectral method, which learns a linear dynamic system (LDS) and is able to discover and exploit general trends across all the notes and short note sequences. The spectral method was rooted in control theory [42] and further developed in the machine-learning field [43]-[45], which has been proved to be both fast and effective in many applications [44][45]. As opposed to maximum likelihood estimation (MLE), the spectral method is a method-of-moments estimator that does not need any random initialization or iterations and also does not suffer from local optima. Very recently, lasso regularization has been incorporated into the learning procedure [46]. We followed this idea and further used group lasso regularization with the spectral method.

2.2.3 One-way ANOVA and within-subject ANOVA

One-way ANOVA can provide a statistical test of whether the means of several groups of data are identical [47][48]. It can be seen as a generalization of the two-sample t-test, which compares the means of two groups. Generally speaking, one-way ANOVA computes an F-test statistic, which is the ratio of variance between groups to the variance within groups. If different group means are close to each other, this F-test statistic will have a relatively low value and hence retain the null hypothesis, showing that the means of different groups are *not* significantly different from each other. On the other hand, if this F-test statistic is greater than a certain threshold, the null hypothesis will be rejected.

We will see one-way ANOVA being used in multiple sections in the thesis to compare subjective ratings on different models. In particular, we used a special kind of ANOVA named within-subject ANOVA [48] (also known as the repeated-measurement study). Within-subject ANOVA can be seen as a generalization of the pair-wise t-test and requires each subject to rate all the models (conditions). The advantage of the within-subject method is that subjects themselves serve as a control variable. Therefore, it can remove the variability due to the individual difference and usually dramatically increase the power of the hypothesis test. In addition, we adopted the method introduced in [49] to compute the mean standard errors (MSEs) and p-values for our within-subjective design and adopted Huynh-Feldt correction [48] when the "sphericity" of our data is not met.

2.3 Music Psychology related work

Most related work in Music Psychology, specifically sensorimotor synchronization (SMS) and entrainment, studies adaptive timing behavior [50]-[62]. Generally, these fields try to discover common performance patterns and high-level descriptive models that could be connected with underlying brain mechanisms. (See Keller's book chapter [61] for a comprehensive overview.) Though the discovered statistics and models are not "generative" from a machine learning perspective and hence cannot be directly adopted to synthesize artificial performances, their musical discoveries and insights inform the design of our features and computational models.

2.3.1 Sensorimotor synchronization (SMS)

SMS [50]-[54] studies how musicians tap or play the piano by following machine generated beats. In most cases, the tempo curve of the machine is pre-defined and the focus is on how humans keep track of different tempo changes. Repp, Keller and Mates argued that adaptive timing requires error correction processes and used a "phase/period correction" model to fit the timing error [51][53]. The experiments showed that the error correction process can be decoupled into period correction (larger scale tempo change) and phase correction (local timing adjustment). This discovery suggests that it is possible to predict timing errors based on musical features on different timing scales.

2.3.2 Entrainment

Compared to SMS, entrainment studies ([55],[56],[58]-[62]) consider more realistic and difficult two-way interactive rhythmic processes. Goebl [62] investigated the influences of audio feedback in a piano duet setting and claims that there exist bidirectional adjustments during full feedback despite the leader/follower instruction. This agrees with the statement on the important role of second piano performance on actively shaping the musical expression in piano duets. Repp [55] did further analysis and discovered that the timing errors are auto-correlated and that how much musicians adapt to each other depends on the music context, such as melody and rhythm. Keller [61] claimed that entrainment not only results in coordination of sounds and movements, but also of mental states. These arguments suggest that it is possible to predict the timing errors (and other musical expressions) by regressions based on different music contexts, and that hidden variables can be introduced to represent mental states.

2.4 Discussion

In conclusion, we have reviewed three fields of related work: Computer Music, Machine Learning & Statistics, and Music Psychology. From Computer Music, we see previous work on generating human-computer interactive performance, expressive music performance, and robot musical gestures, which never informed each other but will be bridged by this thesis. From Machine Learning & Statistics, we see sophisticated computational models and statistical tools, which can be used to generate and evaluate expressive artificial performance. From Music Psychology, we see descriptive models and discovered common patterns of duet performance, which inform us what musical features can be used and how to design our generative models. The three fields of related work, as a joint force, are pointing in the direction of this thesis — generating expressive and collaborative music performance via machine learning.

Chapter 3

Data Collection and Preprocessing

To learn the expressive interaction between musicians, we collect a unique dataset of piano duet performances, where two pianists interact with each other expressively. In this chapter, we first introduce the data collection in Section 3.1. Then in Section 3.2, we present the data preprocessing techniques that detect, inspect, and fix the performance errors.

3.1 Data collection

Musicians: We invited 16 graduate students from the School of Music at Carnegie Mellon University to perform duet pieces in 8 pairs.

Music pieces: We selected 10 pieces of music (see Table 1 for the details) from three books of duet performances [63]-[65] based on their suitable length and difficulty for recording. For all the pieces, the first piano part is a monophonic melody and the second piano part is either monophonic or polyphonic.

Music style: We focus on classical and folk music, whose pitch information is relatively fixed compared with modern (pop, jazz, etc.) music. We further edit the score to eliminate any optional grace notes to standardize the pitch context. A fixed pitch context makes it more straightforward to define *timing* and *dynamics* aspects of expressive music interaction, which is the main focus of this thesis. (Grace notes are considered in another dataset, which will be presented in Chapter 7, to learn improvisational techniques.)

Recording settings: Musicians performed the music by sitting face to face. Pieces were recorded using electronic pianos with MIDI output, therefore all the parameters (dynamics, starting time, ending time, pedal) of every note can be recorded accurately in real time.

Recording procedures: Before the recording session, musicians received the scores about a week in advance; they were asked to practice the pieces individually so they could play their parts fluidly and accurately during the recording session. In the recording sessions, the musicians warmed up and practiced the pieces together for about 10 minutes before the recording began. (So, we did not capture any individual or joint practicing procedure, only the final performance results.) They were instructed to perform the pieces with different interpretations (emotions, tempi, etc.). The first piano player would usually choose the interpretation and was allowed (but not required) to communicate the interpretation with the second piano player before the performance. In each recording session, 1 pair of musicians recorded about 12 to 15 performances of possible different pieces in about 1 hour.

Timeline: The piano duet performance dataset was created over two time periods: Feb – May 2014 and Feb – Apr 2015. For the former, we recorded 3 pieces of music and 5 duet pairs of musicians participated. Each pair performed every piece 7 times, so that we have collected 3 × 5 × 7 = 105 performances. For the latter, we recorded 7 pieces and 3 duet pairs participated. Each pair performed every piece 4 times so that we have collected 7 × 3 × 4 = 84 performances. In total, we have collected 105 + 84 = 189 performances of 10 pieces of music.

Score Preparation: Aside from the performances, the dataset also includes a score version (also in MIDI format) for each piece. Since we only have 10 pieces and the pieces are very short, we manually created the digital score based on the sheet music.

Table 1 shows an overview of the dataset where each row corresponds to a piece of music. The first two columns represent piece index and name. The 3rd to 5th columns represent the number of pairs (of musicians that performed this piece), the number of performances played by each pair, and the total number of performances of each piece of music. The 6th column represents the average performance length. The last column shows whether the 2nd piano part is polyphonic (abbreviated as p.) or monophonic (abbreviated as m.).

Table 1. An overview of the piano duet performance dataset.

index	name	#pair	#perf./pair	#total perf.	avg. length	2^{nd} piano
1	Danny Boy	5	7	35	1’06’’	p.
2	Ashokan Farewell	5	7	35	1’12’’	p.
3	Serenade	5	7	35	1’48’’	p.
4	The Gentle Maiden	3	4	12	1’04’’	p.
5	Berceuse	3	4	12	1’11’’	p.
6	Lament for the Wild Geese	3	4	12	1’13’’	p.
7	La dernière Rose	3	4	12	1’02’’	p.
8	The Sally Gardens	3	4	12	49’’	m.
9	Spanish Love Song	3	4	12	1’36’’	p.
10	The Rose of Tralee	3	4	12	1’04’’	m.

3.1.1 Contribution

Performance datasets with precise labels of (measurements of) control parameters are critical to study expressive music performance. However, such datasets are rare resources compared with raw audio recordings due to the difficulty of measuring musical expression. Currently, most existing labeled datasets focus on classical piano performance. The reasons are twofold. First, piano notes intrinsically have fewer control parameters (only timing, key velocity, and pedal position) compared with other instruments. (For example, the control parameters of violin notes contain bow position, vibrato, etc., which are difficult to measure.) Second, piano control parameters can be fully captured by an electronic or computer-controlled piano. Among the datasets, two famous ones [21][25][66] were created over years by Widmer and his colleagues; one contains 13 Mozart piano sonatas and the other one contains basically the complete Chopin solo piano works. Repp and his colleagues created a dataset [67] that contains multiple performances of 4 piano sonatas. A recently created (and the only open access) dataset is CrestMuse [68], which contains 325 piano performances of 79 pieces.

Our data collection contributes to another important dataset for expressive performance study on top of previous work. The whole dataset is now available online at www.cs.cmu.edu/~gxia/data. As far as we know, it is the first *interactive* performance dataset that contains multiple performances for the each piece of music. The multi-performance nature of the dataset makes it easier to carefully study various aspects of musical expression of every individual note, including timing, dynamics, rolled chord effect, pedal position, etc. The duet nature of the dataset allows us to inspect how musicians interact with each other, studying the expressive performance from both of the piano parts.

3.2 Data preprocessing

Raw performances cannot be directly fed into the computational models. They usually contain errors since musicians will make mistakes by accidentally adding (inserting) and skipping (omitting) notes in performance. In this section, we present the data-preprocessing techniques to deal with performance errors. We first present how to detect such errors by aligning the performances with corresponding scores in Section 3.2.1. Then, we inspect and quantify the errors in Section 3.2.2. Finally, we present how to fix the errors in Section 3.2.3.

3.2.1 Score-performance alignment

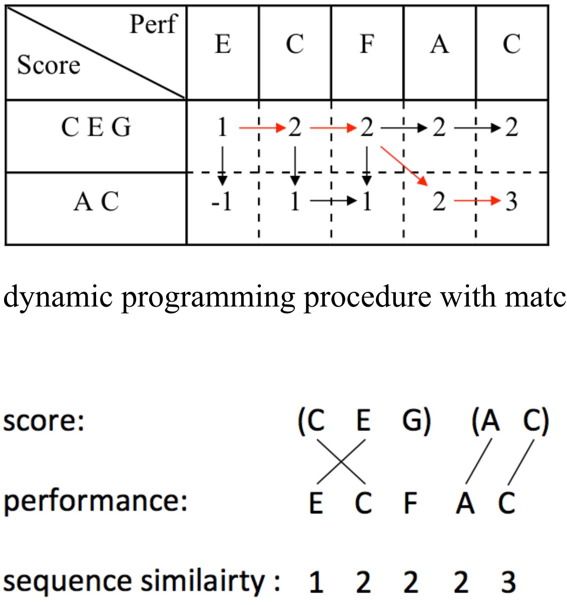
A score can be seen as the expected sequence of notes and chords to be performed. Therefore, we compared the performance with the score to detect the performance errors. Score-performance alignment (comparison) is well studied and more-or-less a solved problem in computer music. Researchers have developed both *online* and *offline* polyphonic alignment algorithms for both *audio* and *symbolic* representations.

For audio-based polyphonic alignment, researchers usually first analyze an audio spectrogram to extract pitch and timing features and then use Viterbi-like methods for the alignment based on extracted features. Cont [11] uses non-negative matrix factorization for polyphonic pitch analysis and then uses a hierarchical hidden Markov model to achieve the alignment by sequential modeling. Raphael [69] introduces a graphical method to detect latent tempo and current position in score.

Compared to audio-based approaches, symbolic alignment is relatively easy since the target files usually already contain accurate pitch and timing information. Bloch and Dannenberg [5] introduce two online algorithms as a part of the first polyphonic computer accompaniment system. Their work uses pitch information and a rating function to find the best fit between performance and score. Hoshishiba et al. [70] propose an offline approach by using dynamic programming and spline interpolation, in which dynamic programming (a discrete version of Viterbi algorithm where transition of emission matrices are manually defined) is used to find the maximum match between performance data and score, and spline interpolation is used to post-process and improve the result.

Our method belongs to offline symbolic alignment since we use MIDI representation and we know the complete performances before the alignment. The alignment is done in two steps: *dynamic-programming* (*DP*) *alignment* and *backward correction.* (An online approach will be presented in Section 5.1.1 for the purpose of real-time score following.)

DP alignment: We slightly modified the method invented by Bloch and Dannenberg [5] for the DP alignment step. Generally speaking, it uses a dynamic matching algorithm that takes a performance as sequential inputs and matches the notes one-by-one to the score. At each step of the alignment, it computes the best association between performance and score based on the associations of the previous step. We measured the best association via a *sequence similarity*, which is computed as the number of matched score notes minus the number of skipped score notes. As in [5], added notes do not decrease the sequence similarity.



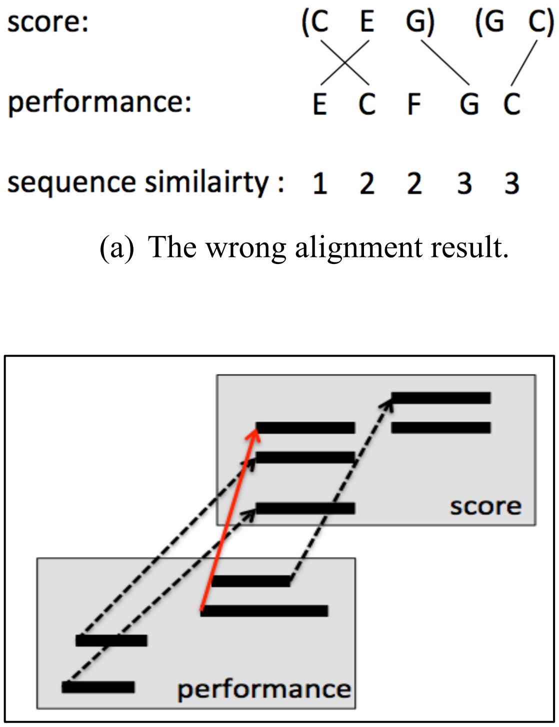
(a) The dynamic programming procedure with matching paths.

(b) The alignment result.

Figure 3. An illustration of DP alignment (with one added and one skipped notes).

Figure 3 shows a simple example of the DP alignment, in which the score contains two chords and the performance has one added note and one skipped note. Subfigure (a) shows the dynamic programming procedure with the matching paths. The first row represents the performed note sequence, the first column represents the score (a sequence of chords), the numbers represent the sequence similarities, and the arrows represent the matching paths. The paths are interpreted as follows: a vertical one means to skip notes, a diagonal one means to perform a new chord, a horizontal one means either to perform an extra note that is not in the score (if the sequence similarity does not increase) or to continue performing the current chord (if the sequence similarity increases). After the matrix is filled, the algorithm picks up the largest number at the last row and traces back the paths to figure out the alignment. Subfigure (b) shows the alignment result, where the first line is the score with notes belonging to the same chords in parentheses, second line is the performed note sequence, and the third line is the sequence similarity associated with each performed note. Note that we did not use the traditional concept of "substitution" in string matching; for the DP alignment task a substituted note is equivalent to one omitted note plus one added note.

Backward correction: The DP alignment algorithm performs correctly in most cases for the performances in our dataset. However, a problem is that it only considers the order of notes and ignores the exact timings. As a consequence, it may systematically cause a mismatch when adjacent chords share the same note. Figure 4 shows an example of the mismatch, where the score and performance are very similar to the example in Figure 3, with only one note difference. Here, subfigure (a) shows the wrong alignment result and subfigure (b) shows a piano roll representation where we can better see which note belongs to which chord from the timing information.



(a) The wrong alignment result.

(b) A piano roll representation. Dotted arrows represent correct matches while the solid arrow represents the false match.

Figure 4. An illustration of mismatch when two adjacent chords share the same note.

In this case, the top note G in the 1st chord is skipped in the performance and the next chord's 1st performed note happens to share the same pitch with the skipped note. As a consequence, the 1st chord "borrows" the missing note from the performance of the 2nd chord. In the worst case, if all the chords share the same note, this mismatch behavior could happen recursively. To address this problem, the backward correction algorithm starts from the last chord and recursively recovers the borrowed notes, if any.

3.2.2 Inspecting performance errors

The score-performance alignment algorithm can tell us which score notes are skipped by the performance and which performed notes are not in the score. To be specific, we identify and quantify three categories of performance errors: *insertion, note omission,* and *compound-event omission*.

- **Insertion:** To perform notes that are not in the score. We measure it by the *insertion rate*, which is computed as the ratio between the number of inserted notes and the number of total notes in score.
- **Note omission:** To skip score notes during the performance. We measure it by the *note omission rate*, which is computed as the ratio between the number of omitted notes and the number of total notes in score.
- **Compound-event omission:** To skip an entire compound event during the performance. A compound event is either a single note whose onset is unique in the score or a chord (a set of notes sharing the same onset in score). We measure such omissions by *compound-event omission rate,* which is computed as the ratio between the number of omitted compound events and the number of total compound events in score.

It is important to notice that for the first piano, compound-event omission is equivalent to note omission since the first piano performance is monophonic. For the second piano, a compound-event omission is considered a more severe mistake than a note omission; we will soon see in Section 3.2.3 that such mistakes are more difficult to fix.

Table 2 shows an overview of the performance errors for both of the piano parts of all the pieces, where we see the average insertion rate, note omission rate, and compound omission rate (abbreviated as c.e. omission rate) across different performances. In the last row, we highlight the overall average error rates, where we see that all three kinds of error rates are under 2%. Such rates are very similar to (and even smaller than some of) the reported numbers associated with the famous datasets [21][66][67]. The low error rates indicate that the quality of our dataset is high, at least from the perspective of performance correctness.

Table 2. An overview of the performance errors.

index	part	insertion rate (%)	note omission rate (%)	c.e. omission rate (%)
1	1st piano	0.45	0.08	0.08
	2nd piano	1.17	1.26	0.40
2	1st piano	0.33	0.03	0.03
	2nd piano	1.49	1.96	1.75
3	1st piano	0.15	0.49	0.49
	2nd piano	2.01	7.26	0.90
4	1st piano	0	0.10	0.10
	2nd piano	1.38	1.77	1.16
5	1st piano	0.10	0.10	0.10
	2nd piano	2.29	3.48	1.45
6	1st piano	0.34	0	0
	2nd piano	0.64	1	1.05
7	1st piano	0.45	0.27	0.27
	2nd piano	0.76	1.56	0.09
8	1st piano	0	0	0
	2nd piano	0.09	0.35	0.35
9	1st piano	0.18	0.18	0.18
	2nd piano	0.45	0.55	0.35
10	1st piano	0	0	0
	2nd piano	0.14	0	0
total	1st piano	**0.19**	**0.16**	**0.16**
	2nd piano	**1.04**	**1.92**	**0.75**

3.2.3 Fixing performance errors

After detecting the performance errors, we fix them. Performances with no errors are comparable on every individual note, which enables more straightforward learning and evaluation procedures. According to the three different error categories listed in Section 3.2.2, we use three different strategies to fix the error notes as follows:

- **To fix inserted notes**: We simply throw out any inserted notes as they are not part of the score.
- **To fix omitted notes within a chord:** If a note is omitted but at least one note in the corresponding chord is performed, we use the performed notes in the chord as the *recovery context* to fix that omitted note. To recover the *onset* and *dynamic* of a note, we adopt the mean of the onsets and dynamics of the performed notes in recovery context. To recover the *duration* of a note, we rely on the ratio between performance duration and score IOI (inter-onset-interval). We first take the mean of this ratio of the notes in recovery context, and then recover the actual duration by multiplying the corresponding score IOI. With onset, dynamic, and duration, an omitted note can be fully recovered (re-synthesized).
- **To fix an entire omitted compound event:** We consider a larger-scale *recovery context*, which is the first piano notes within 2 beats of the omitted compound event. To recover the onset, we use linear tempo estimation. Figure 5 shows an example, where the omitted chord's score time is 9 and its recovered performance onset is *x*. In this case, the recovery context is the first piano notes from 7th to 11th beat, the "+" signs represent the onsets of the first piano notes within the context, and the dotted line is the tempo map computed via linear regression. To recover the dynamic and duration-to-IOI ratio, we again adopt the mean of the notes within the recovery context.

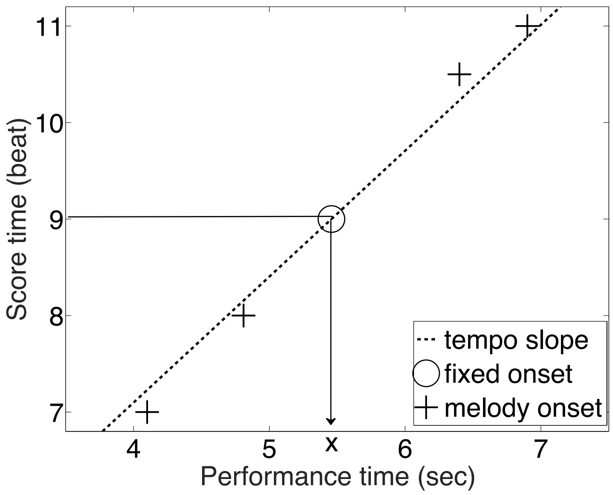


Figure 5. An illustration to recover the onset of an omitted compound event.

Chapter 4

Feature Representations

Mathematically, piano performance can be considered a function of frequency (pitch) and loudness (dynamics) over time. For our collected data in MIDI format, each note is encoded by its pitch, dynamics, onset (starting time), duration, and corresponding pedal movements. To be specific, pitches are integers in semitones with the middle C being 60, dynamics are integers in velocities units (speed with which the keys are hit), and timings are floating point numbers in seconds. We only used the sustain pedal on our electronic piano. (Though the pedal can affect resonance of an acoustic piano, its function is limited to extend the duration of notes on our electronic piano until the pedal is released, even if the keys are released.)

In this chapter, we describe the features we use as input and output for our learning system, i.e., what measurable high-level abstractions of duet performance we consider in addition to the properties of individual notes in order to learn the expressive interaction of the two piano parts. The designed feature representation serves as an intermediate layer between the data (introduced in the last chapter) and the computational models (to be presented in the next chapter).

In particular, since the final goal is to generate an expressive artificial 2nd piano performance given a score and a human 1st piano performance, we identify *input features* and *output features*. The input features are designed to represent the score and the 1st piano performance, while the output features are designed to represent the 2nd piano performance. We first present the output features in Section 4.1 since they are more concise and easy to understand than the input features. Then, we present the input features in Section 4.2. Since the designed input features are very rich, we only show an abstract design scheme and leave the actual input features to be presented in the next chapter with each individual computational model, which generates the output given the input.

4.1 Output features

As in most expressive performance studies, we focus on the prediction and generation of starting time and dynamics since they are the two most fundamental features for piano performance. Also, we chose to use a *compound event* as an output unit. (Remember that a compound event is either a single note whose score onset is unique or a chord, defined as a set of notes that begin simultaneously in the score.) The reason is that the performance decision of a chord is more of an indivisible whole than independent controls on each note.

Thus, for each compound event, the output feature is a 2-dimensional vector, with the first dimension being the starting time and the second dimension being the dynamics. One question naturally rises: how do we derive the onset and dynamics of a chord based on a real performance where notes do not really begin simultaneously or have exactly the same dynamics? One approach is to imagine a performance in which chord notes are truly simultaneous and dynamics are equal. We would like to find such a performance that is perceptually very similar to the actual performance. In other words, for each specific chord in the actual performance, what onset time and dynamic level can functionally replace the onsets and dynamics of the performed notes in that chord?

Very few works consider the onset and dynamics differences within chords in part due to a lack of theoretical foundations. As a consequence, when dealing with chords, existing *expressive performance* studies [26][71] usually either stick to the melody or rely on the parameters of the first or the highest/lowest note, even though authors realize this is not an appropriate solution. In this section, we present a more systematic study, where chord onset is explored in Section 4.1.1 and chord dynamics are explored in Section 4.1.2. In the end, we show that compared with using the onset/dynamics of a certain note in a chord, a *mid-range* onset/dynamics is a better choice. The mid-range onset/dynamics refers to the average between the minimum and maximum onset/dynamics of the notes within a chord. Parts of the experiments in this section were contributed by Mutian Fu and published in a co-authored paper [72] and in Ms. Fu's master's thesis [73].

4.1.1 Chord performance onset

Research question: If we replace all the note onsets of a chord by a single onset time, where should we place this time to let the chord sound most like the original chord?

Model: We define a chord's *onset interval* as the timing difference between its first and last onsets. Figure 6 shows an example, where we see the distributions of the onset interval for each chord in the piece *Danny Boy* over 35 performances. Here, we see that most onset intervals are very small (less than 30 milliseconds) but a few can be more 100 milliseconds. The smaller ones are considered unintentional asynchronies of fingers, while the bigger ones are considered intentional rolling or arpeggiations (to perform notes in chords sequentially, usually from lowest to highest in pitch.)

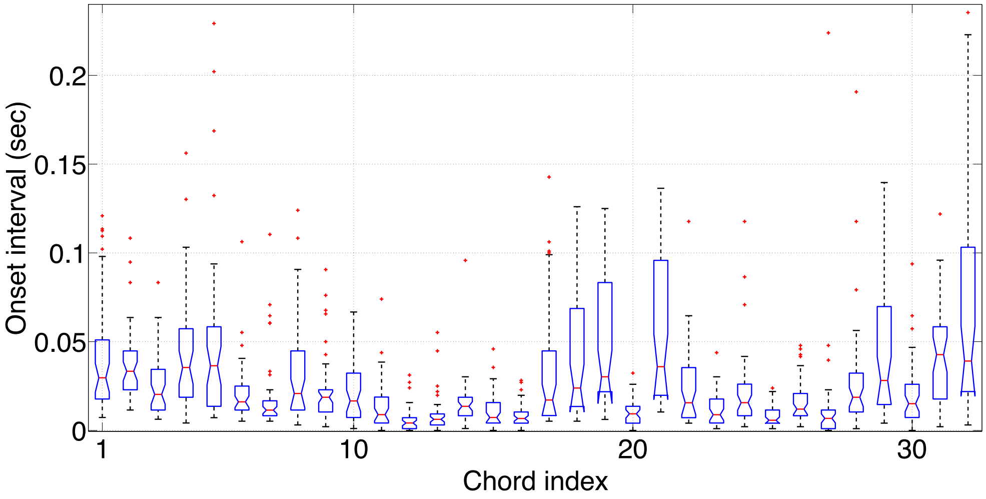


Figure 6. A boxplot of the onset intervals of the chords in Danny Boy. (Danny Boy contains 90 compound events and 32 of them are chords.)

We define the *chord onset time* or *chord onset* as the single onset time that is most representative of the overall chord. We will discover later that the *chord onset* is within the range of the onset times of notes belonging to the chords and has some relationship with them. In particular, we used a *ratio model*:

$$t(r) = (1 - r)t_{min} + r \cdot t_{max} \tag{1}$$

Here, t_{min} and t_{max} refer to the first and last note onsets in a chord, respectively. (Onset interval is equal to $t_{max} - t_{min}$.) The parameter r characterizes the relative location of chord onset. According to the value of r, the chord onset can be located in three different conditions:

- $r < 0$: Before the first onset of the chord.
- $0 \leq r \leq 1$: Between first onset and the last onset of the chord.
- $r > 1$: After the last onset of the chord.

Objective measurement: If the local tempo around a chord is stable, the ground truth chord onset can be linearly estimated from neighboring melody onsets of the first piano performance. We can use this estimated value to help find the optimized *r* value. Of course, performance tempo is not always stable, but this method can at least give us a rough ground truth of chord onsets.

To be specific, we consider the melody notes within 2 beats of a chord and transfer the chord performance onset estimation problem into a tempo estimation problem. Formally, if a chord's score onset is denoted as *s* and the estimated performance onset is denoted by *t'*, we estimate *t'* based on the melody notes whose score onsets are within the range of $[s-2, s+2]$. First, we compute a linear mapping between performance onsets and score onsets of the melody notes within this range. Then, if we denote the slope and the intercept of this linear mapping as α and β, respectively, the *t'* is estimated by:

$$t' = \alpha \cdot s + \beta \qquad \textbf{(2)}$$

This process is illustrated by Figure 7, in which the '+' symbols represent the melody notes, and the circle symbols represent notes belonging to the chord. The line represents the tempo map computed by linear mapping, and the star point represents the estimated chord performance onset. After the *t'* of every chord in a piece is estimated, we find the optimal *r* value by minimizing the absolute error:

$$\hat{r} = \underset{r}{\operatorname{argmin}} \sum_{\text{perf}} \sum_{\text{chord}} |t' - t(r)| \qquad \textbf{(3)}$$

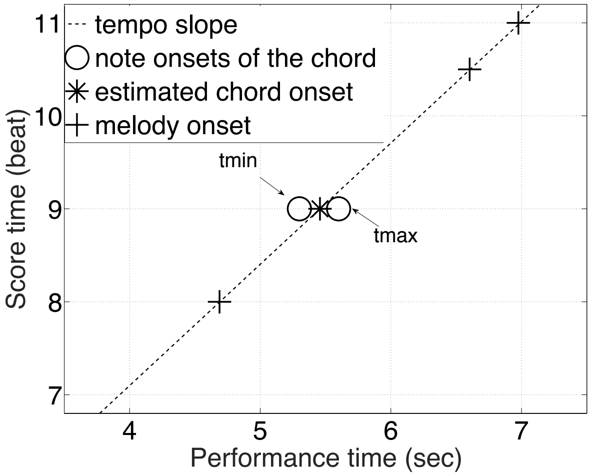


Figure 7. An illustration of chord performance onset estimation by local tempo estimation.

We used the first three pieces of music, which contain more total chords, to compute the optimal *r* value. Figure 8 shows the results, where we see that the optimal *r* values are all within the range from 0 to 1, indicating that the chord onset consistently lies within the range of chord onset intervals. Here, each curve corresponds to a piece of music, the *x*-axis represents the ratio parameter *r*, and the *y*-axis represents the total absolute difference between t' and $t(r)$. Therefore, small numbers indicate better results. The optimal values are 0.42 for *Danny Boy*, 0.13 for *Ashokan Farewell*, and 0.78 for *Serenade*.

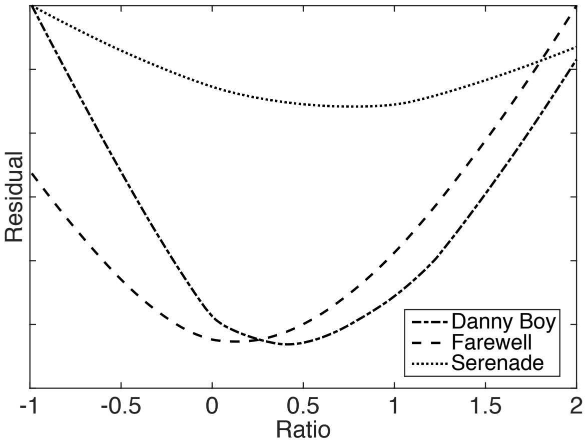


Figure 8. Objective measurement results of the ratio model.

Subjective measurement: A more convincing way to figure out chord onsets is to subjectively rate the similarities between the original chords and the synthesized chords corresponding to difference models.

We designed a double-blind online survey, where we selected three segments by clipping them from our performance dataset. Since the timing differences are subtle and the experiment requires careful listening, each segment is about 10 seconds long. In addition, the selected segments meet the following three standards:

- They contain no performance errors.
- Each segment contains at least 5 chords.
- The chords in the segments cover onset intervals in a wide range, but avoid extremely small (near zeros) or large (>.5 second) onset intervals.

During the survey, for each selected segment, the subjects first listened to the original human performance and then listened to *four* synthesized versions. (The order was randomized both within segments for different synthesized versions and across the three segments.) Referring to the notations in equation (**1**) and (**2**), the four synthesized performance versions were:

- $t(r = 0)$: A synthesized chord onset lies on t_{min}, which is the onset of the first (and usually the lowest) note in the original chord.
- $t(r = 0.5)$: A synthesized chord onset lies on $(t_{min} + t_{max})/2$, which is the mid-range of the onset interval.
- $t(r = 1)$: A synthesized chord onset lies on t_{max}, which is the onset of the last note (usually also the highest note in pitch) in the original chord.
- t': A synthesized chord onset lies on the estimated timing using linear tempo estimation based on neighboring melody notes.

It is important to notice that notes in a synthesized chord share exactly the same onset (the chord performance onset). For other parameters (dynamics and durations), the synthesized notes keep the same as in the original chord. After listening to each synthesized version, subjects rated the similarities between the original and synthesized performances using a 5-point Likert scale, from 1 (very low similarity) to 5 (very high similarity).

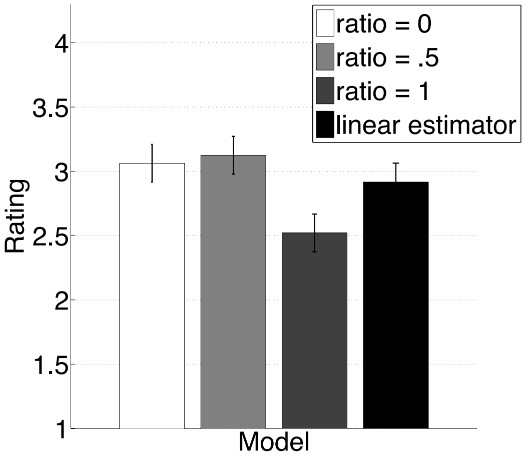


Figure 9. Subjective measurement results of chord onsets. (Higher is better.)

A total of n = 24 subjects completed the survey. The aggregated result is shown in Figure 9, where we see that the *mid-range* onset (r = 0.5) has the highest rating out of the four choices. Here, different colors represent different methods to compute the chord onset, the heights of the bars represent the means of the ratings, and the error bars represent the mean standard errors. The difference between r = 0 and r = 0.5 is not statistically significant (with p-value larger than 0.05) but nevertheless we see r = 1 is a bad choice. If we have to go with one chord performance onset, we choose r = 0.5 because its result is marginally better.

4.1.2 Chord performance dynamics

Research question: If we replace all the note dynamics of a chord by a single chord dynamic, how loud should it be to let it sound most like the original chord?

Model: Similar to the chord performance onset problem, we used a *ratio model*:

$$d(r) = (1 - r)d_{min} + r \cdot d_{max} \qquad \textbf{(4)}$$

where r is the ratio parameter, d_{min} is the smallest note dynamics in a chord, and d_{max} is the largest note dynamics in a chord.

Subjective measurement: Unlike the chord onset problem, we only used subjective measurement to figure out chord dynamics. (The dynamics of the two piano parts in duet performance can sometime differ a lot so it is less reliable to estimate chord dynamics based on nearby melody notes.) To design the survey, we selected two segments with no performance errors. Again, each segment is about 10 seconds long and contains more than 5 chords.

The design of the survey was almost the same as the chord onset problem, except that we only had three synthesized performance versions due to the lack of objective measurement. Notes in a synthesized chord share the same dynamic (i.e., the chord performance dynamics apply to each note in the respective chord) and their other parameters keep the same as in the original chord. The three synthesized versions were:

- $d(r = 0)$: The synthesized chord dynamic is d_{min}, which is the dynamic of the softest note in the original chord.

- $d(r = 0.5)$: The synthesized chord dynamic is $(d_{min} + d_{max})/2$, the mid-range of the note dynamics.
- $d(r = 1)$: A synthesized chord dynamic is d_{max}, which is the dynamic of the loudest note in the original chord.

The same group of subjects participated the survey as for the chord onset problem. The aggregated result is shown in Figure 10. We again see that the *mid-range* choice (*r=0*) has the highest rating on average. In addition, the difference between the three choices of r value is more significant (with p-value smaller than 0.005) compared with the chord dynamics problem.

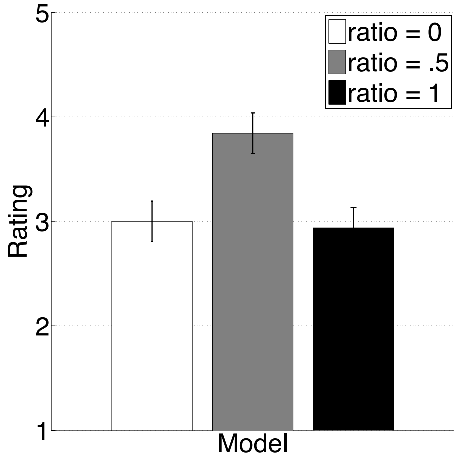


Figure 10. Subjective measurement results of chord onsets.

4.1.3 Summary

In summary, we assume the unit for the output feature (what we want to predict and generate) is a compound event. If a compound event is a single note, its output features are just the onset and dynamics of that note. If a compound event is a chord, the output features for each note within the chord are the chord's *mid-range* onset and dynamics, which are computed as the average between the minimum and maximum onset/dynamics of the notes belonging to the chord. Our experiments have shown that the "mid-range choice" outperforms both the "first note choice" and "highest note choice" used in most expressive performance research.

Note that the output features do not include chord duration, pedal movements, and the parameters of individual notes within a chord. We leave the study of these parameters for future work. For individual notes' parameters within a chord, our study in this section suggests that future work can use the chord onset/dynamics as the anchor points (instead of the first or highest note) and consider the onset/dynamics interval as an important parameter to recover the parameters of individual notes.

4.2 Input features

The input features are designed to represent various aspects of the score and the 1st piano performance. Unlike the output features that focus only on individual compound events, the input features consider a much richer music context. In particular, features are designed from four aspects of expressive duet performance, as shown in Figure 11.

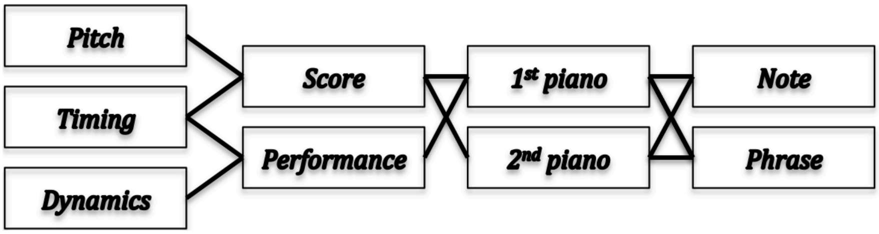


Figure 11. The designed input feature scheme.

Here, each column represents an aspect of duet performance, and each line between the columns represents a possible interaction. For example, the top "Pitch-Score-1st piano-Note" path represents the score pitch of the 1st piano at a note level. Note that there is no edge between the "Pitch" block and the "Performance" block since pitches are defined in the score. Also, the "Performance" block does not link to "2nd piano" block since the 2nd piano performance is the output.

Although the parameters of individual notes are simple, complex musical expression can be revealed when the context becomes richer. For example, a "Dynamics-Performance-1st piano-Phrase" feature can reveal a crescendo of 1st piano performance, a "Timing-Score-1st piano-Note" feature can reveal a local rubato, and a "Pitch-Score-2nd piano-Phrase" feature can reveal the pitch contour of the second piano. Here, we only show the abstract design scheme for the input features. The actual input feature lists will be presented soon in the next chapter on computational models, where we will see a series of models, each of which adopts a subset of the graph paths in Figure 11 to exact the corresponding input features.

Chapter 5

Computational Models

Different function approximations are designed to model the relationship between one pianist's musical expression and another's. We first introduce the baseline model, which assumes that local musical expression is steady. For the advanced models, we start from very low-dimensional representations and local models that that operate note-by-note, and gradually progress to high-dimensional representations and more general models that can apply to the whole piece of music.

Based on the learned models, an artificial performer will be able to generate (decode) its own musical expression by interacting with a human pianist. As stated in Chapter 4, for piano notes, musical expression is encoded by timing and dynamics (i.e., once we know these parameters, we can re-synthesize the notes) and we consider onset time and dynamics as they are the two most important features for piano performance.

In this chapter, we will present five models in turn. Experimental results associated with each model are shown right after the model descriptions. (A global view of the results of all models can be found in Section 5.6.) We consider human performance as the *ground truth* and use the absolute difference [74] between machine prediction and ground truth as the objective measurement for prediction accuracy. Therefore, small numbers mean better predictions. (Note that human performance varies, too, and we will look at human variability later.) To compare the results of different methods and to choose the optimal parameters, we use cross-validation. Since we are interested in music performance with a limited number of rehearsals, we often sample a small subset of the training samples. To avoid over fitting, we also exclude the rehearsals performed by the *same* pair of performers. (The effect of the performer will be discussed in detail in the next chapter). We show both detailed results (over score time) and high-level statistics. Since the results are very consistent over the 10 pieces of music, we present only detailed results for one piece, *Danny Boy*.

A discussion is held after each model's experimental results to show the strengths and limitations of the model. The discussion part also reveals some important discoveries and the motivation for developing the next-level model.

5.1 Baseline approach

The baseline model is known as *score following and automatic accompaniment*, often briefly named automatic accompaniment. This model has been used in human-computer interactive performance for over 30 years. A system diagram of the standard automatic accompaniment system is shown in Figure 12. The system takes a human's monophonic piano performance (the 1st piano) as input and outputs the performance of a polyphonic collaborative part (the 2nd piano).

We first describe the standard procedures of the baseline approach, score matching and tempo estimation, from Section 5.1.1 to 5.1.3. Then we introduce dynamics estimation, which is less studied in previous works, in Section 5.1.4. Finally, we present the experimental results and motivate advanced models in Section 5.1.5 and 5.1.6.

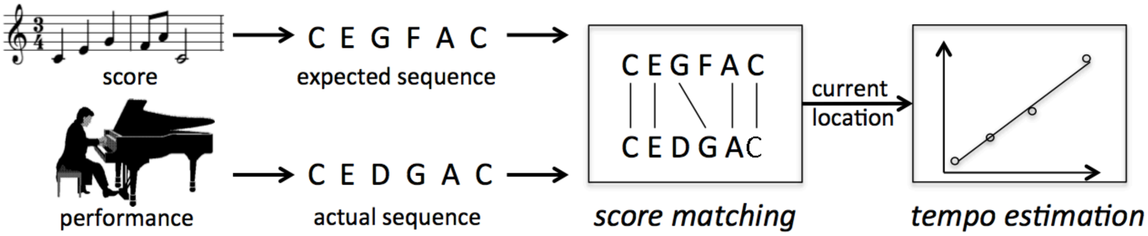


Figure 12. The system diagram for score following and automatic accompaniment.

5.1.1 Score matching

Given the human performance, the first step of the baseline approach is score matching, which keeps track of the current score location by finding the best match between score and performance. In our case, both score and performance are represented by a sequence of pitch symbols, with performance being the actual sequence and the score being the expected sequence. If the performance exactly follows the order of the score, we would simply update score locations stepwise. However, since human performers will make mistakes by accidentally inserting and skipping notes, we need an online matching algorithm. The current system adopts the solution introduced in [2], which first computes the "sequence similarity" associated with each performance note and then updates the score location *only* when this length exceeds previously reported ones. Formally, the sequence similarity is computed by:

$$SequenceSimilarity = \#\text{ matched note} - \#\text{ skipped note} \qquad \mathbf{(5)}$$

Here, # represents the number of elements. Figure 13 shows an example, where the first line is the score, second line is the performance, and the third line is the sequence similarity associated with each performed note.

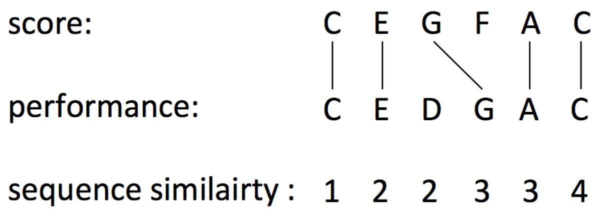


Figure 13. An illustration of score following with one added and one skipped notes.

In this example, the algorithm reports a match and updates the score location after C, E, G, A, C are performed, since their sequence similarities exceed previous ones. For the performed note D, the matched length does not increase because it is not in the score. For the performed note A, though the number of matched note increases by 1, this increment is offset by the skip of score note F. From the perspective of string edit distance, this matching algorithm has zero unit cost for an insertion (added note such as D) and one unit cost for a deletion (skipped note such as F).

5.1.2 Tempo estimation

Given the matching results of score following, tempo estimation quantifies how fast/slow the human performance is against the timings specified in the score. This result will be used later to schedule the performance of the 2nd piano. To be specific, we adopt a "performance-reference timing" 2-D representation (as shown in Figure 14) and represent tempi as slopes in this 2-D plane. Here, *reference* usually refers to a score. The unit of score time is *beat*, the unit of performance time is *second,* and hence the unit of tempo is *beats per second*.

Formally, let the matched notes reported by score following be $m = [m_1, m_2,\ldots, m_i,\ldots]$. Also, let the corresponding performance time and score time be $p = [p_1, p_2,\ldots, p_i,\ldots]$ and $s = [s_1, s_2,\ldots, s_i,\ldots]$, respectively. Then, the tempo is defined as $v = [v_1, v_2,\ldots, v_i,\ldots]$. If there are n matched notes within the score time interval of $[s_i - dur, s_i]$, v_i is computed via the method of least squares:

$$v_i = \begin{cases} \frac{\sum_{k=i-n+1}^{i}(p_j-\bar{p})(s_j-\bar{s})}{\sum_{k=i-n+1}^{i}(p_j-\bar{p})}, & n > 1 \\ 1, & n = 1 \end{cases} \qquad (6)$$

Here, *dur* is a parameter often set to be 4 beats, and:

$$\bar{p} = \frac{1}{n}\sum_{k=i-n+1}^{i} p_k \tag{7}$$

$$\bar{s} = \frac{1}{n}\sum_{k=i-n+1}^{i} s_k \tag{8}$$

Figure 14 shows an example of tempo estimation corresponding to the score following example in Figure 13, where the solid line represents the tempo of the last matched note C. Note that we do not estimate the tempi of unmatched notes.

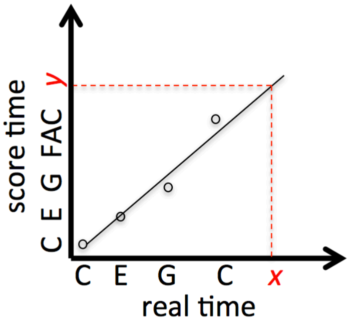


Figure 14. An illustration of tempo estimation.

Based on the tempo estimation, the timing prediction for the 2nd piano is computed by extrapolating the tempo (slope). Note that the 2nd piano also has a pre-defined score that is synchronized with the score for human performance. Figure 14 shows an example, where the nearest note of the 2nd piano after the last human performed note C is at beat *y* and therefore its actual performance time will be at time *x* (in seconds). Formally, let *s'* = [s_1', s_2',..., s_i',...] be the score time of the notes of the 2nd piano, and let s_j' be the score time of the 2nd piano right after s_i. Then, its corresponding performance time p_j' is estimated by:

$$\widehat{p_j}' = \frac{s_j' - v_0}{v_i} \tag{9}$$

where

$$v_o = \bar{s} - v_i \cdot \bar{p} \tag{10}$$

5.1.3 Median performance for tempo estimation

A simple modification of the tempo estimation (introduced in 5.1.2) is to change the reference from score timing into statistics of the timing in rehearsals. This idea was first introduced by Vercoe [75] but not well evaluated in subsequent studies. Nevertheless, the rationale is inspiring: just taking the score time as the reference is essentially "sight reading"; performance timing in rehearsals can be a better reference for tempo estimation.

To be specific, we introduce the notion of a *median performance*. A median performance is constructed note by note, by taking the median tempo of each note across multiple rehearsals. (Remember that timings of missed notes were reconstructed in Section 3.2.3.) In other words, a median performance is a basic memorization of the performance history.

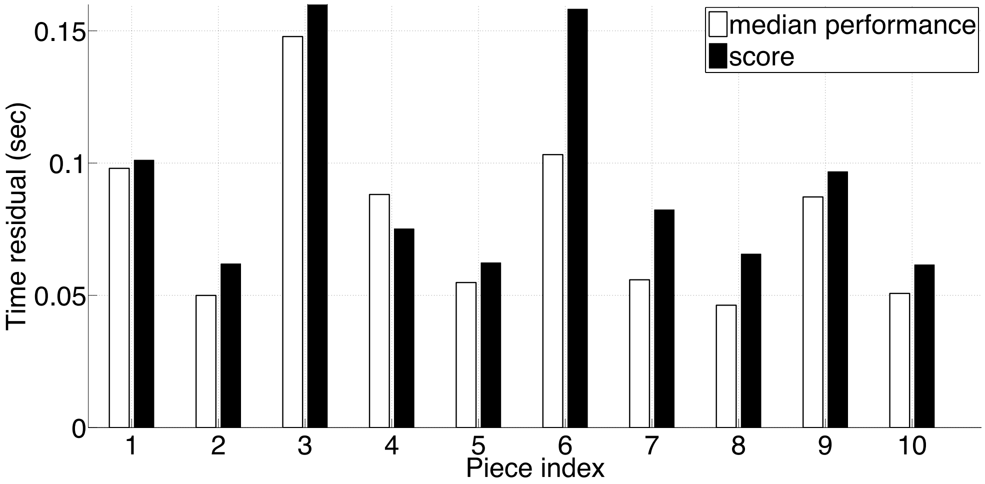
5.1.4 Dynamics estimation

Unlike the timing feature, dynamics is not defined in the score in detail. In fact, the estimation of dynamics is rarely mentioned in previous work. However, we can at least estimate a dynamic level by looking at the performance of the 1st piano. By assuming that the dynamics of a piece is locally stable, for each note of the 2nd piano, we use the average dynamics over several 1st piano notes right before it as our baseline prediction. Formally, let the dynamics of the matched notes of the 1st piano be $d = [d_1, d_2, \ldots, d_i, \ldots]$ and let the dynamics of the notes of the 2nd piano be $d' = [d_1', d_2', \ldots, d_i', \ldots]$. Similar to Section 5.1.2, if there are n matched notes within the score time interval of $[s_i - dur, s_i]$, the dynamics of the note of the 2nd piano right after s_i is estimated by:

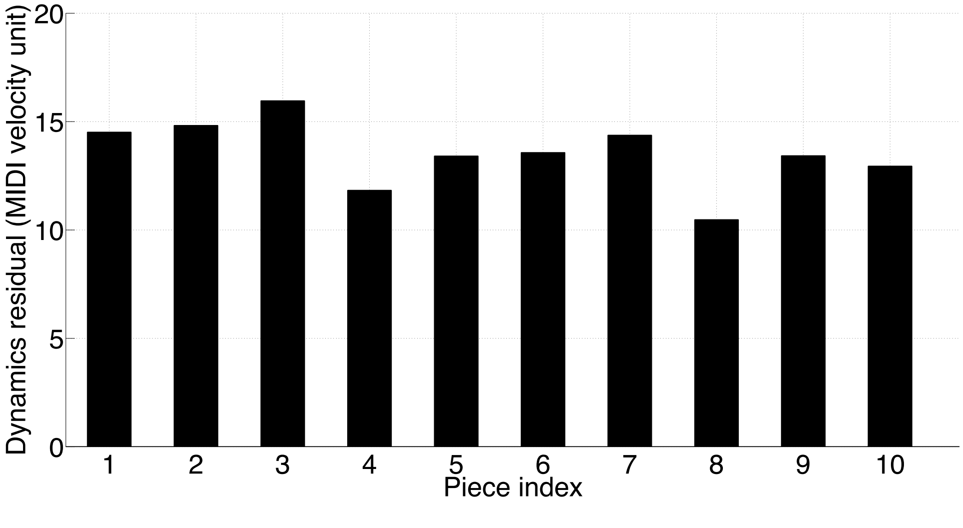
$$\widehat{d_j}' = \bar{d} = \frac{1}{n}\sum_{k=i-n+1}^{i} d_k \qquad \textbf{(11)}$$

5.1.5 Results

Figure 15 shows the cross-validation result of the baseline approach, where we see that the timing residuals range from 50 milliseconds to 150 milliseconds and the dynamics residuals range from 10 to 15 MIDI velocity units. For both subfigures, the *x* axis is the piece index and the *y* axis is the mean absolute residual. Subfigure (a) shows the timing results, where the black bars represent using the score as references, while the white bars represent using the median performances as references. Subfigure (b) shows the dynamics results. (Note that using score and median performance as reference yield same result for dynamics prediction.)



(a) A global view of starting time result.



(b) A global view of dynamics result

Figure 15. The cross-validation results of the baseline approach. (Smaller is better.)

5.1.6 Discussion and new prediction features

We see that the baseline model is very straightforward. It assumes locally steady tempo and dynamics. In other words, it takes local averages as the *trends* but ignores the *deviation* between the trend and actual timing/dynamics. In contrast to the baseline model, advanced models care about such deviations and aim to predict them by discovering their regularities over the course of a performance from rehearsals. To some extent, measured deviations are just a manifestation of inconsistent performance, i.e. performance errors. However, we will see that a significant portion of these deviations is predictable, indicating that expressive timing and dynamics have a systematic component in addition to whatever randomness is present.

To be precise, for advanced models (Section 5.2 – Section 5.5), we define *expressive timing* and *expressive dynamics* as the deviation of the actual performance from the *trend* as predicted by the baseline model. Formally, by referring to the notations in Section 5.1.2 and 5.1.3, these two features are defined as:

$$\mathrm{ExpTiming_j} \overset{\mathrm{def}}{=} p_j{}' - \widehat{p_j}' \qquad \textbf{(12)}$$

$$\mathrm{ExpDynamics_j} \overset{\mathrm{def}}{=} d_j{}' - \widehat{d_j}' \qquad \textbf{(13)}$$

It is important to notice that for advanced models, we do not directly predict the raw timing and dynamics but rather predict the *expressive* features defined above. In other words, advanced models are built upon the baseline model — we first *de-trend* the performance of the 2nd piano using the baseline model (score is the reference) and then predict the *residuals* using more advanced models.

5.2 Note-specific approach

The note-specific approach assumes that expressive timing/dynamics of the notes is linearly correlated and learns a different model for each individual note. Intuitively, when musicians slow down/speed up and become louder/softer, we assume that there are certain patterns of expressive features that can be characterized by linear regression. Therefore, if we have observed enough performance examples of every note, a note's expressive features can be estimated based on the expressive features of previous notes.

5.2.1 Model

By referring to the general feature scheme described in Section 4.2, we use the features corresponding to the "Timing-Score-1st piano-Note" path. Let $X = [x_1, x_2, \dots, x_N]$ be the expressive timing/dynamics of the notes played by the 1st pianist; let $Y = [y_1, y_2, \dots, y_M]$ be the expressive timing/dynamics of the notes played by the 2nd pianist. (*N* and *M* are note indices). Then the model is:

$$y_i = \beta_0^{y_i} + \sum_{j=1}^{p} \beta_j^{y_i} x_{\mathrm{over}(y_i)-j} \tag{14}$$

Here, p is the lag parameter and $x_{over(y_i)-j}$ are the p note times in *X* previous to y_i. Thus, $over(y_i)$ is the smallest index of the element of *X* whose score time is greater or equal to the score time corresponding to y_i. For example, in Figure 16, let the 1st and 2nd systems be the score for the 1st and 2nd piano, respectively. If the note in the dotted circle corresponds to y_i and the lag parameter p is equal to 3, the notes in the circle would corresponds to $x_{over(y_i)-j}$.

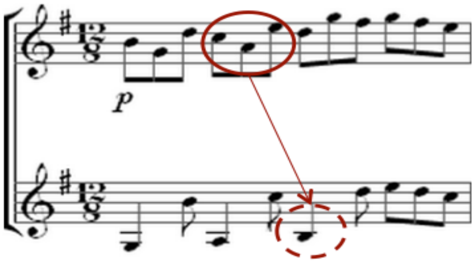


Figure 16. An illustration of the note-specific approach.

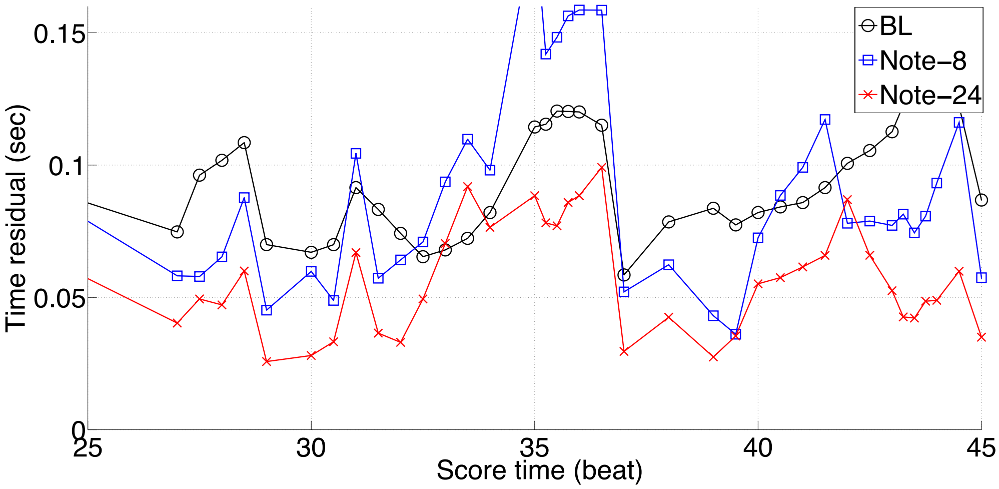
It is important to notice that the note-specific approach trains a different set of parameters for each note, which is reflected by the superscript of β. The advantage of this approach is that each note gets a tailored solution, while the disadvantage is that many training rehearsals are needed because there are so many parameters to estimate.

5.2.2 Results

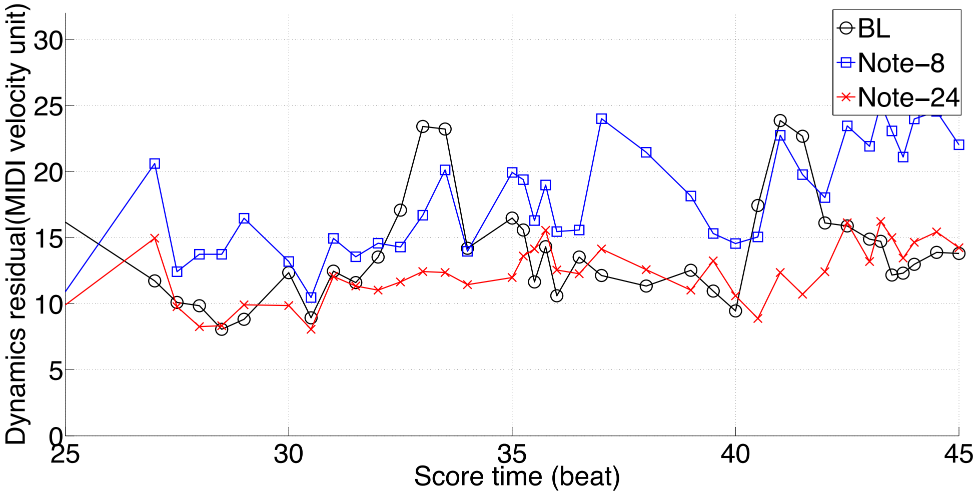
Figure 17 shows the cross-validation results of the note-specific approach, where we see that the note-specific approach outperforms the baseline when there are a lot of rehearsals. Here, subfigures (a) and (b) are the detailed results for one piece of music, and (c) and (d) are the overall results for all pieces of music. Through cross-validation, we adopt the lag parameter *p* to be 3 or 4. To have a fair comparison, we adopt the baseline results using the median performances as references.

For (a) and (b), the *x* axis is the score time and the *y* axis the residual. The curve with circle markers represents the baseline method, the curve with square markers represents the note-specific method trained by 8 rehearsals, and the curve with "x" markers represents the note-specific method trained by 24 rehearsals. We can see that the note-specific method works very well when there are a lot of training rehearsals but not so well when the training size is reduced to 8.

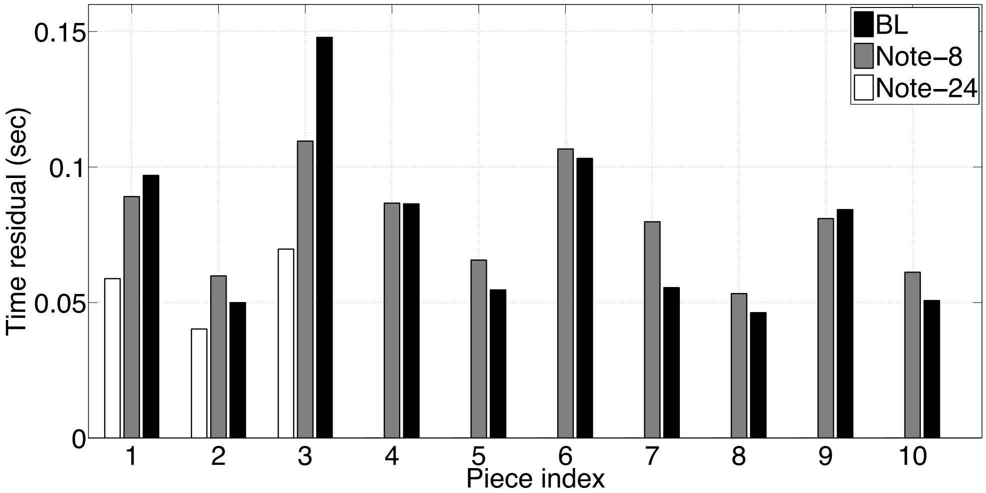
For (c) and (d), the *x* axis is the piece index and the *y* axis is the residual. The black bars represent the baseline method, the grey bars represent the note-specific method trained by 8 rehearsals, and the white bars represent the note-specific method trained by 24 rehearsals. Note that we only see white bars for the first three pieces because there are not enough rehearsals for other pieces. We see that the global results are consistent with the detailed results — the note specific approach works well given lots of rehearsals but almost always worse than the baseline when the training set size shrinks to 8.



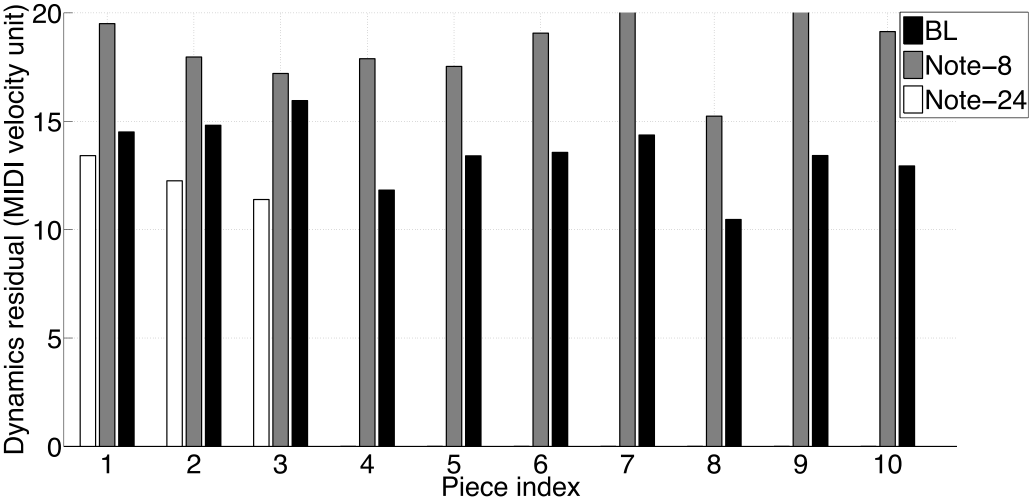
(a) A detailed view of starting time results.



(b) A detailed view of dynamics results.



(c) A global view of starting time results.



(d) A global view of dynamics results.

Figure 17. The cross-validation results of the note-specific approach. (Smaller is better.)

5.2.3 Discussion

The note-specific approach outperforms the baseline when trained on 24 rehearsals. This result suggests that there exists a local relationship of musical expression; even when just looking ahead for 3 to 4 notes (the lag parameter *p* in Section 5.2.1), we can still capture some regularities of musical expression using local linear regressions. This result reveals the 1st discovery.

Discovery 1: *There exist some local patterns of musical expression beyond what the baseline captures that can be captured by linear regression*.

In fact, this local linear pattern is more obvious on the expressive features introduced in Section 5.1.5. When we use raw timing/dynamics features, the note-specific model does not perform as well as using expressive features. This result indicates that *de-trending* performance by the score improves the model's performance:

Discovery 2: *De-trending performance leads to better prediction, at least for the local linear model.*

Last but not least, when we shrink the training set size to 8, the note-specific approach is not as effective. This deficiency is because for a 4 to 5 dimensional (*p*+1) linear regression, an 8-sample training set easily results in over-fitting. Though 24 rehearsals is doable, it is considered a very large number in practice. We need a more general method to shrink the training set size while still outperforming the baseline. Having some notes sharing their parameters may solve this problem, hence our next-level model: the rhythm-specific approach.

5.3 Rhythm-specific approach

To improve the generality of the model, the rhythm-specific approach introduces an extra dummy variable to encode the score rhythm context of each note. Intuitively, we assume that notes of the same rhythm context share the same pattern of the expressive features. This is mathematically equivalent to training a different set of parameters for each rhythm context, rather than for each note as in the last section.

5.3.1 Model

Formally, let *X* and *Y* be the same as in the note-specific approach. The rhythm-specific model is then:

$$y_i = \beta_0^{\mathrm{rhythm}(y_i,q)} + \sum_{j=1}^{p} \beta_j^{\mathrm{rhythm}(y_i,q)} x_{\mathrm{over}(y_i)-j} \tag{15}$$

where $\mathrm{rhythm}(y_i, q)$ is the categorical variable representing the score rhythm context of the note y_i within *q* notes. To be more precise, the rhythm context of y_i is defined as the inter-onset-intervals of the q 1st piano's notes right before y_i. As *q* increases, the possible values of $rhythm(y_i, q)$ will also increase. For example, in Figure 18, again let the 1st and 2nd systems be the scores for the 1st and 2nd piano, respectively. When *q* is equal to 3, the two notes in the dotted circles would share the same $\mathrm{rhythm}(y_i, q)$. The two notes' rhythm contexts are shown by the circled notes.

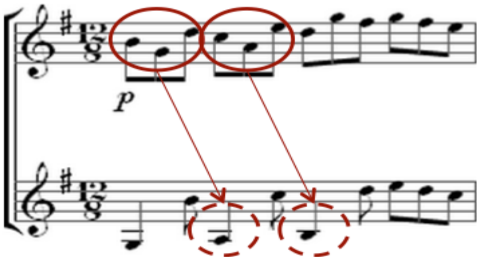


Figure 18. An illustration of the rhythm-specific approach.

It is important to notice that many notes share the same rhythm context within a piece of music and hence share the same set of parameters. As a consequence, the model can gain more information from each rehearsal, and fewer training rehearsals are needed compared to the note-specific approach. For example, if *N* notes share the same rhythm context, we would just need *1/N* rehearsals for these *N* notes in order to gain the same amount of training data compared with the note-specific approach. We refer to this number associated with each note as *rhythm frequency*. However, this improvement does not apply to some "odd notes" whose rhythm contexts are unique (rhythm frequency equal to 1). For these notes, the rhythm-specific approach reduces to the note-specific approach.

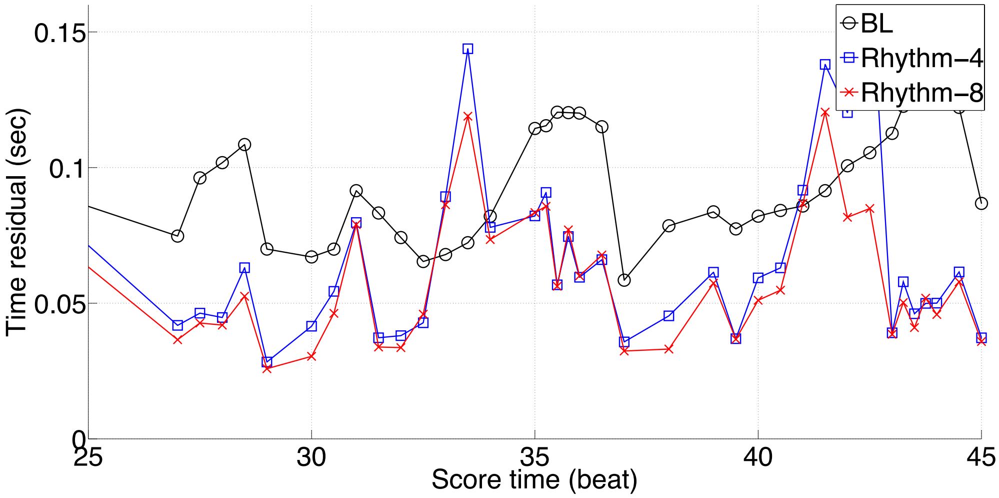
5.3.2 Results

Figure 19 shows the result of the rhythm-specific approach, where we see better results than the baseline when the training set size is 8. Through cross-validation, the lag parameter *p* and rhythm context parameter *q* are both set to be 3 or 4. Here, subfigures (a) to (c) show the detailed results, while (d) and (e) show the overall results.

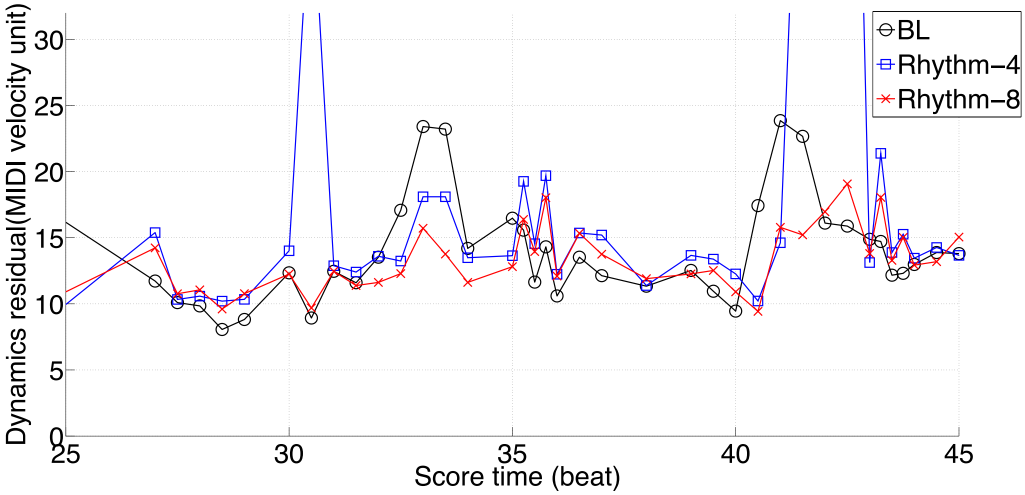
From (a) and (b), we see that when there are 8 training rehearsals, the rhythm-specific method improves the performance compared to the note-specific method and is able to outperform the baseline. Here, the curve with diamond markers represents the baseline method, the curve with square markers represents the rhythm-specific method trained by 4 rehearsals, and the curve with "x" markers represents the rhythm-specific method trained by 8 rehearsals. Subfigure (c) shows the rhythm frequency, where we see two "odd notes" around the 41st beat.

However, when we shrink the training size to 4, the "odd notes" are not predicted well and in fact the result is off the scale shown here. In addition, the bad performance for dynamics prediction around the 30th beat also corresponds to small rhythm frequency.

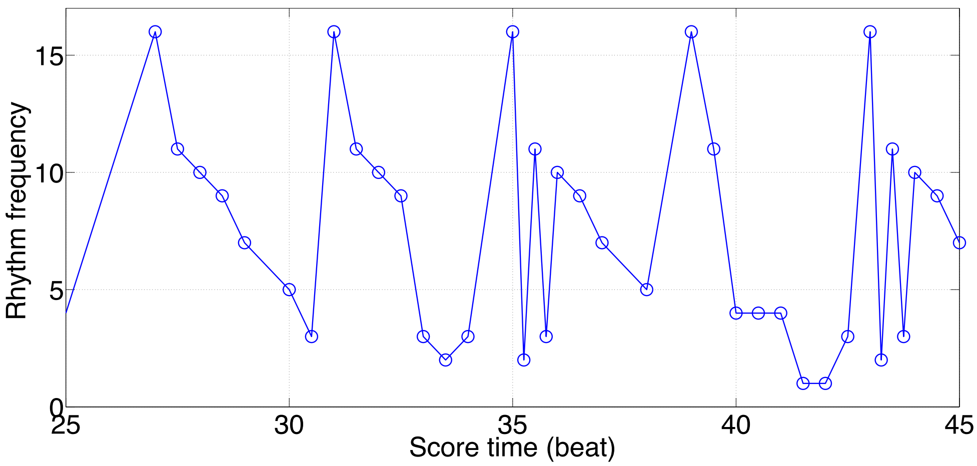
From (d) and (e), we again see that the results are consistent across different pieces — 8 rehearsals is almost always enough for a better result compared with the baseline while 4 rehearsals is not enough for most pieces. Here, the *x* axis is the piece index and the *y* axis is the residual. The black bars represent the baseline method, the grey bars represent the rhythm-specific method trained by 4 rehearsals, and the white bars represent the rhythm-specific method trained by 8 rehearsals.



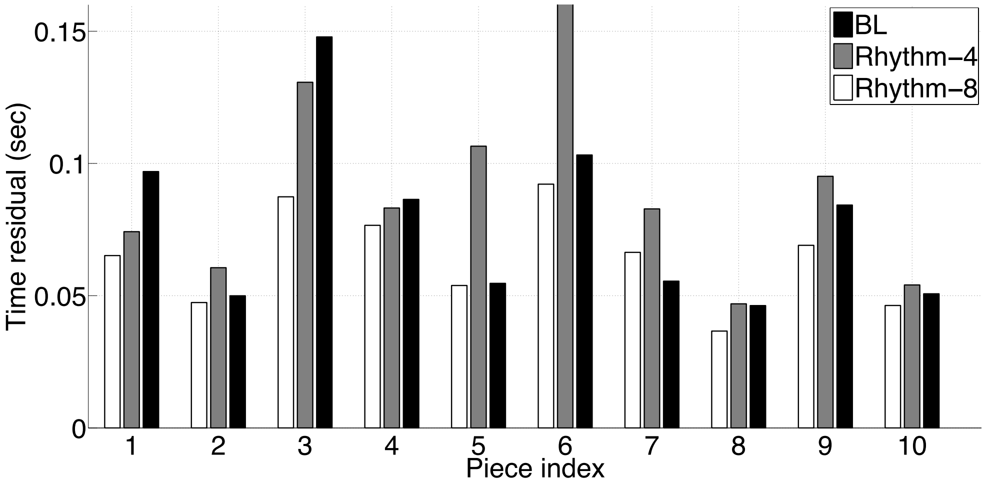
(a) A detailed view of starting time results.



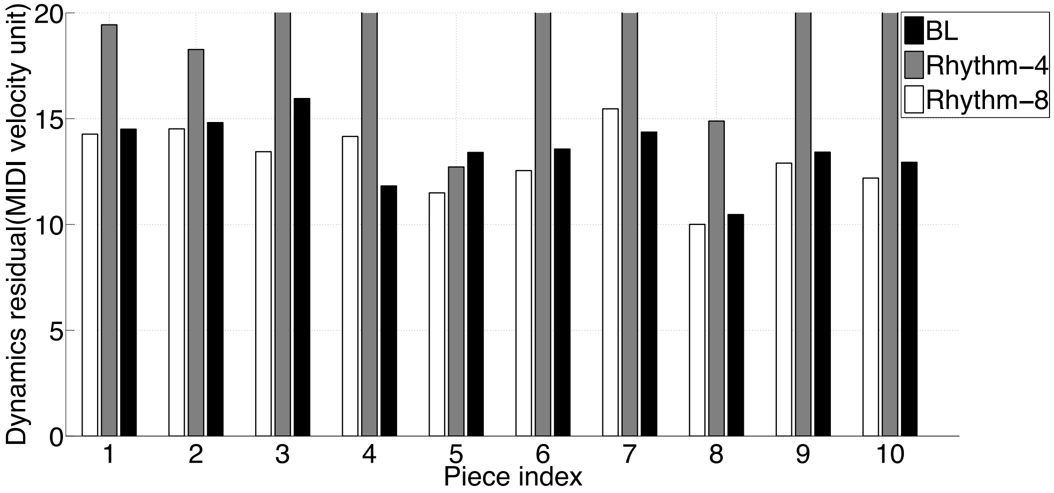
(b) A detailed view of dynamics results.



(c) A detailed view of rhythm frequency.



(d) A global view of starting time results.



(e) A global view of dynamics results.

Figure 19. The cross-validation results of the rhythm-specific approach. (Smaller is better.)

5.3.3 Discussion

When trained on 8 rehearsals, the rhythm-specific approach improves greatly compared with the note-specific approach. This improvement proves our assumption stated at the beginning of Section 5.3. (I.e., notes of the same rhythm context follow similar patterns of expressive features.) In fact, we also experimented with gathering randomly or according to other criteria, e.g., pitch contour, down beat/up beat. However, these approaches do not improve the performance at all; only gathering the notes according to rhythm contexts outperforms the baseline:

Discovery 3: *Notes of the same rhythm context share similar local linear relationships of expressive features.*

Another observation is that the timing results for the rhythm-specific approach are more accurate than the dynamics result. Though the dynamics prediction is more accurate compared with the note-specific approach, its advantage over the baseline is less obvious. In other words, gathering the notes according to rhythm context is a better strategy for expressive timing prediction.

Discovery 4: *Expressive timing is related more to the rhythm context than expressive dynamics.*

Last but not least, if we compare the residuals carefully, the rhythm-specific approach trained on 8 rehearsals does *not* outperform the note-specific approach trained on 24 rehearsals. However, this result does not mean that the gain of training set size, i.e., the rhythm frequency is less than *24/8 = 3*. Figure 20 shows the average rhythm frequencies associated with each piece of music, where this number is larger than 3 for all pieces. This finding indicates that rhythm context is informative for expressive features (especially expressive timing), yet notes of the same rhythm context certainly do *not* follow exactly the same pattern of musical expression. In order to gain better prediction and further shrink the training set size, merely using the rhythm context is not enough. This motivates the next-level model, the general-feature approach.

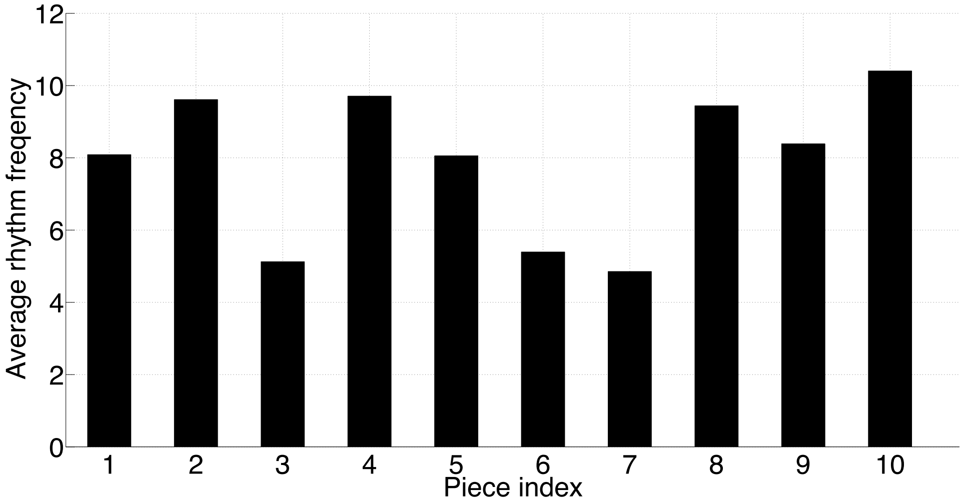


Figure 20. A global view of the average rhythm frequency of each piece of music.

5.4 General-feature approach

To further improve the model's generality and predict the expressive features by more than rhythm context, a more general and comprehensive representation is designed. Features are designed by going through all the possible paths according to the feature scheme introduced in Section 4.2.

5.4.1 Model

In this section, we use $U = [u_1, u_2, \ldots, u_M]$ to denote the general features (dependent variables excluded) of the notes of the 2nd piano and let $Y = [y_1, y_2, \ldots, y_M]$ be our target feature as in Section 5.2 and 5.3. The relationship between *U* and *Y* is modeled by:

$$Y = BU \tag{16}$$

where *Y* is 1-by-M, B is 1-by-P, and *U* is P-by-M. P is the dimensionality of the feature space. This equation can be solved easily by computing the Moore-Penrose pseudo-inverse.

As an alternative, we also consider a *group lasso* [40] penalty, which is to find the optimal parameters by solving:

$$\min_{B \in R^P} \left(\|Y - BU\|_2^2 + \lambda \sum_{l=1}^{L} \sqrt{p_l} \|B_l\|_2 \right) \tag{17}$$

The advantage of group lasso regularization is that it not only reduces the burden for training but also tries to discover the dominant aspect of duet piano performance that could be used to predict the expressive features. In this equation, λ is the penalty parameter, *l* is the *feature group* index, p_l is the dimensionality of l^{th} feature group, and B_l is the parameters corresponding to the l^{th} feature group. A feature group is a meaningful subset of *U*, whose coefficients are weighted equally.

5.4.2 Feature representation

Now let's take a close look at the feature groups of *U*. To formally represent u_t, we introduce an auxiliary notation $R = [r_1, r_2, \ldots, r_M]$ to denote the raw score information and musical expression of the 1st piano and describe the mapping from *R* to each component of u_t. To be more specific, the feature groups are:

High Pitch Contour: For the chords within a certain time window up to and including *t*, extract the highest-pitch notes and fit the pitches by a quadratic curve. Then, *high pitch contour* for *t* is defined as the coefficients of the curve. Formally:

$$\widehat{\beta_t}^{\text{high}} \overset{\text{def}}{=} \underset{\beta}{\operatorname{argmin}} \sum_{i=0}^{p} \left(r_{t-p+i}^{\text{highpitch}} - quad_\beta(t-p+i) \right)^2 \tag{18}$$

Here, p is a context length parameter and $quad_\beta$ is the quadratic function parameterized by β.

Low Pitch Contour: Similar to *high pitch contour*, we compute $\widehat{\beta_t}^{\text{low}}$ for *low pitch contour*. Formally,

$$\widehat{\beta_t}^{\text{low}} \stackrel{\text{def}}{=} \underset{\beta}{\operatorname{argmin}} \sum_{i=0}^{p} \left(r_{t-p+i}^{\text{lowpitch}} - quad_\beta(t-p+i) \right)^2 \tag{19}$$

Beat Phase: The relative location of *t* within a measure. Formally:

$$\text{BeatPhase}_t \stackrel{\text{def}}{=} (t \; mod \; \text{MeasureLen})/\text{MeasureLen} \tag{20}$$

Phrase Phase: The relative location of *t* within a phrase. Formally:

$$\text{PhrasePhase}_t \stackrel{\text{def}}{=} (t \; mod \; \text{PhraseLen})/\text{PhraseLen} \tag{21}$$

Inter-onset-interval (IOI) Context: IOIs of the *p* closest notes directly before *t*. Formally:

$$\text{IOIContext}_t \stackrel{\text{def}}{=} \left[r_{t-p}^{\text{IOI}}, r_{t-p+1}^{\text{IOI}}, \cdots, r_{t-1}^{\text{IOI}} \right]^T \tag{22}$$

Here, IOI is the score time distance from the onset of one note to the onset of the next note.

Tempo Context: Tempi of the *p* closest notes directly before *t.* This is a timing feature on a relatively large time scale. Formally:

$$\text{TempoContext}_t \stackrel{\text{def}}{=} \left[r_{t-p}^{\text{Tempo}}, r_{t-p+1}^{\text{Tempo}}, \cdots, r_{t-1}^{\text{Tempo}} \right]^T \tag{23}$$

Here, the tempo of a note is defined as the slope of the least-squares linear regression between the performance onsets and the score onsets of the preceding notes with *dur* beats. This feature is very similar to the concept of *v* in equation (**6**) introduced in Section 5.1.2. The difference is that here we use multiple intervals to do tempo estimation and concatenate the features together.

Expressive Timing Context: A description of how much the *p* closest notes' onsets deviate from their tempo curves. Compared to the tempo context, this is a timing feature on a relatively small scale. Formally:

$$\mathrm{ExpTimingContext}_t \overset{\mathrm{def}}{=} \left[r_{t-p}^{\mathrm{Exptiming}}, r_{t-p+1}^{\mathrm{Exptiming}}, \cdots, r_{t-1}^{\mathrm{Exptiming}}\right]^T \tag{24}$$

Remember that the expressive timing of each note was defined in equation (**12**). Here we also use multiple values for *dur* and concatenate the features together.

Dynamic Context: MIDI velocities of the *p* closest notes directly before *t.* Formally:

$$\mathrm{DynamicContext}_{\,t} \overset{\mathrm{def}}{=} \left[r_{t-p}^{\mathrm{Vel}}, r_{t-p+1}^{\mathrm{Vel}}, \cdots, r_{t-1}^{\mathrm{Vel}}\right]^T \tag{25}$$

Duration Ratio Context: the duration ratio of the *p* closest notes directly before *t.* Formally:

$$\mathrm{DurRatioContext}_{\,t} \overset{\mathrm{def}}{=} \left[r_{t-p}^{\mathrm{DurRatio}}, r_{t-p+1}^{\mathrm{DurRatio}}, \cdots, r_{t-1}^{\mathrm{DurRatio}}\right]^T \tag{26}$$

Here, the duration ration is defined as the ratio between a note's duration and its IOI.

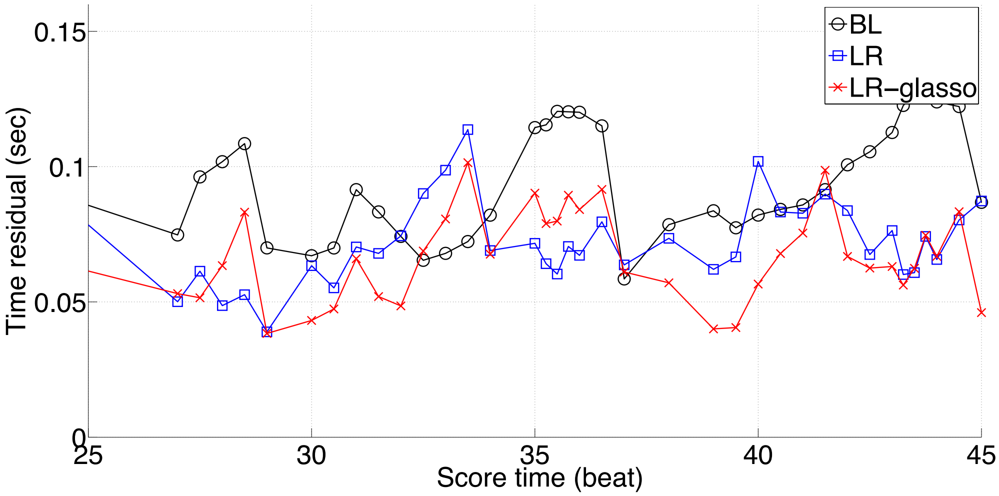
The input feature (independent variable), u_t, is a concatenation of the above features. (We have also tried other features and mappings, e.g., meter and down beat, and finally picked the ones above through experimentation.) Depending on different *p* parameters, the input feature has around 60 to 100 dimensions in our experiments below.

5.4.3 Results

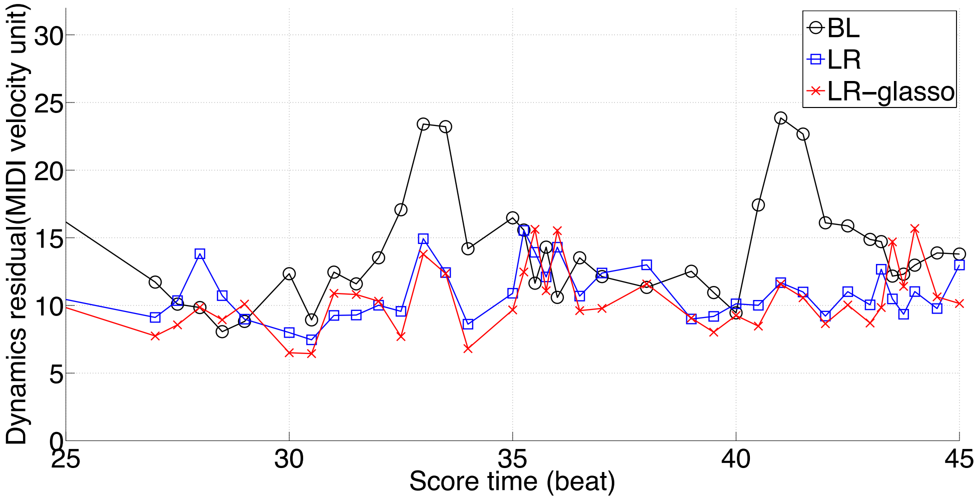
Figure 21 shows the cross-validation results of the general-feature approach, where we finally start to see better results than the baseline trained on only 4 rehearsals. Again, (a) and (b) show the detailed results while (c) and (d) show the overall results.

From (a) and (b), we see that with the regularization, the general-feature approach outperforms the baseline almost everywhere across the piece of music. Here, the curve with circle markers represents the baseline method, the curve with square markers represents the raw linear regression method, and the curve with "x" markers represents group lasso penalty.

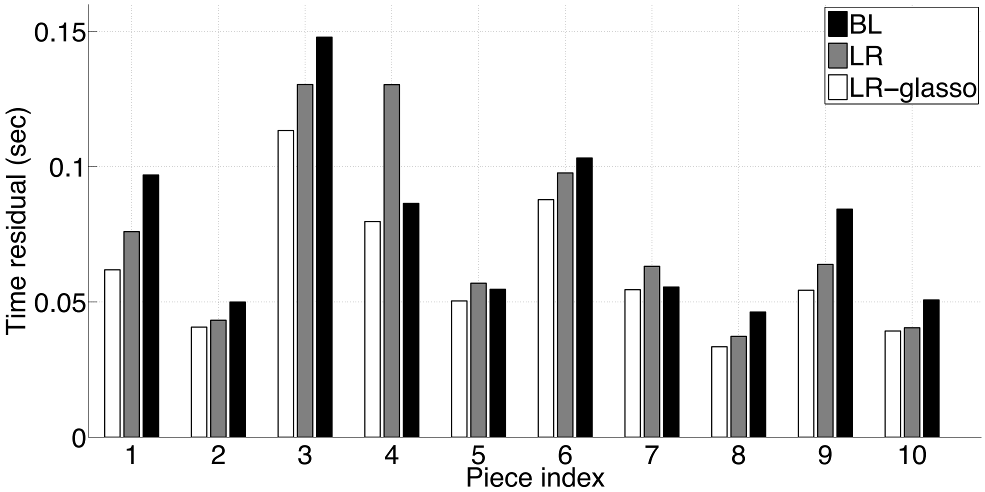
For (c) and (d), the black bars represent the baseline method, the grey bars represent the raw linear regression, and the white bars represent the group lasso penalty. Again, we see that the results are consistent across different pieces — 4 rehearsals are now enough for most pieces to have a better prediction than the baseline. We also see that the group lasso penalty yields slightly better result than raw linear regression. The improvements are more obvious for timing prediction than dynamics prediction.



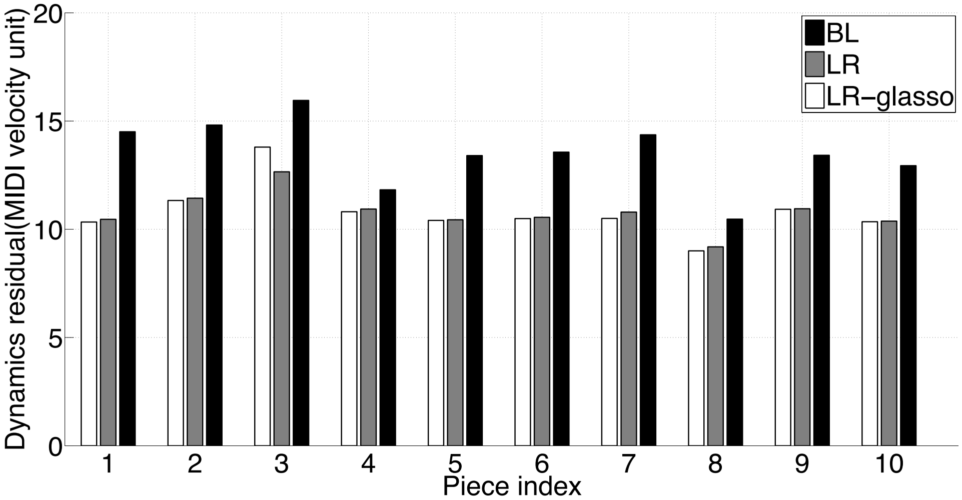
(a) A detailed view of starting time results.



(b) A detailed view of dynamics results.



(c) A global view of starting time results.



(d) A global view of dynamics results.

Figure 21. The cross-validation results of the general feature approach trained on 4 rehearsals. (Smaller is better.)

5.4.4 Discussion

When the training set size is 4, we see a dramatic improvement with the general-feature approach compared with the rhythm-specific approach. For both starting time and dynamics prediction, the general-feature approach outperforms the baseline. This improvement indicates that the designed general-feature scheme introduced in Section 4.2 and the corresponding features proposed in Section 5.4.2 capture useful information from music context in performance. It is important to notice that almost all of the features are very local; no feature captures information beyond the scope of a phrase. In conjunction with the discovery in Section 5.2.3, the result from the general-feature approach reveals a deeper insight:

Discovery 5: *When looking ahead for a phrase, linear regression can capture musical expression across different notes based on only 4 rehearsals.*

Another important discovery comes from the regularization. By checking the coefficients after the group lasso regularization, we see that only *expressive timing context*, *tempo context*, and *IOI context* are retained for timing prediction, while all the features are retained for dynamics prediction. This suggests that we only need a subset of the features to predict expressive timing but need all of them to predict expressive dynamics. Remember that we have already discovered the tight bond between rhythm context feature and expressive timing in Section 5.3.3. Now, we have a clearer view:

Discovery 6: *Expressive timing is mainly related to the timing aspects of music context while expressive dynamics is related to all aspects of music context.*

Last but not least, though we see better results than the baseline when trained on a very practical number of rehearsals, the notes are still predicted independently. In other words, the prediction of one note does not affect the prediction of its subsequence notes. This weakness motivates the next-level model, the linear dynamic system approach.

5.5 Linear dynamic system approach

We finally consider the time-series effect, linking up the notes and modeling them by a linear dynamic system (LDS). In particular, we assume there exist some low dimensional hidden mental states. The mental states change smoothly over time and control the expressive features of the 2nd piano. Intuitively, the LDS approach can be seen as adding another regularization to the expressive features by adjacent notes' musical expression.

5.5.1 Model

Formally, we adopt the following graphical representation:

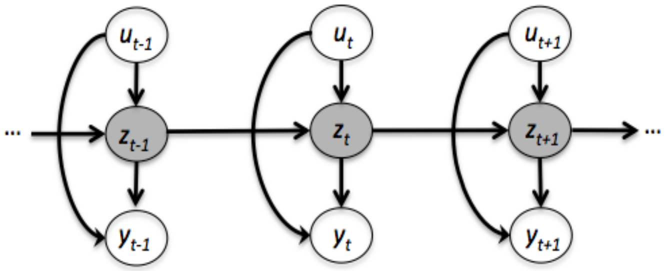


Figure 22. The graphical representation of the LDS, in which grey nodes represent hidden variables.

In Figure 22, *u* and *y* are very similar to the representations as in the Section 5.4. In LDS, *y* is referred to as *observation*, *u* is referred to as *input*, and *z* is referred to as the *hidden state.* The evolution of this time series can be described by the following equations:

$$z_t = Az_{t-1} + Bu_t + w_t \quad w_t \sim \mathcal{N}(0, Q) \tag{27}$$

$$y_t = Cz_t + Du_t + v_t \qquad v_t \sim \mathcal{N}(0, R) \tag{28}$$

Here, $y_t \in \mathbb{R}^2$ and its two dimensions correspond to expressive timing and dynamics, respectively, $u_t \in \mathbb{R}^P$ is a much higher dimensional vector, and $z_t \in \mathbb{R}^n$ is a relatively lower dimensional vector. *A, B, C,* and *D* are the main parameters of the LDS. Once they are learned, we can predict the performance of the 2nd piano based on the performance of the 1st piano.

5.5.2 Performance sampling

Before introducing the learning method, let's look at an important difference between the features used in LDS approach and in the general feature approach (Section

5.4). In Section 5.4, we did not consider the time-series effect and the subscript *t* for *y* refers to note index. For the LDS approach, one question arises: should *t* represent note index or score time? Inspired by Todd's work on algorithmic composition [76], we assume that musical expression evolves with score time rather than note indices, and therefore define *t* as score time in this section. Since music notes have different durations, we "sample" the performed notes (of both the 1st piano and the 2nd piano) at the resolution of a half beat, as shown in Figure 23.

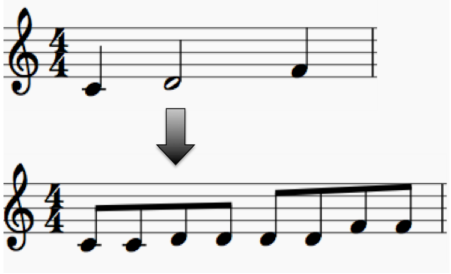


Figure 23. An illustration of performance sampling.

To be more specific, if a note's starting time aligns with a half beat and its *inter-onset-interval* (IOI) is equal to or greater than one beat, we replace the note by a series of eighth notes, each having the same pitch, dynamic, and duration-to-IOI ratio as the original note. If a note's staring time does not align with a half beat (e.g. a sixteenth note), we simply do not consider this note in the learning process but use linear interpolation to recover its timing and dynamics based on its neighbor notes for prediction. Note that we still play the notes as originally written; the sampled representation is only for learning and prediction.

5.5.3 Spectral learning procedure

To learn the model, we use a spectral method, which is rooted in control theory [42] and then further developed in the machine learning field [43]-[45]. Spectral methods have proved to be both fast and effective in many applications [43][45]. Generally speaking, a spectral method learns hidden states by predicting the performance future from features of the past, but forcing this prediction to go through a low-rank bottleneck. In this section, we present the main learning procedure with some underlying intuitions, using the notation of Section 5.5.1.

Step 0: Hankel matrices construction.

We learn the model in parallel for fast computation. In order to describe the learning procedure more concisely, we need some auxiliary notation. For any time series $S = [s_1, s_2, \dots, s_T]$, the "history" and "future" Hankel matrices are defined as follows:

$$S_H \overset{\text{def}}{=} \begin{pmatrix} s_1 & \cdots & s_{T-d} \\ \vdots & \ddots & \vdots \\ s_{\frac{d}{2}} & \cdots & s_{T-\frac{d}{2}-1} \end{pmatrix}, S_F \overset{\text{def}}{=} \begin{pmatrix} s_{\frac{d}{2}+1} & \cdots & s_{T-\frac{d}{2}} \\ \vdots & \ddots & \vdots \\ s_d & \cdots & s_{T-1} \end{pmatrix} \tag{29}$$

Also, the "one-step-extended future" and "one-step-shifted future" Hankel matrices are defined as follows:

$$S_F^+ \overset{\text{def}}{=} \begin{pmatrix} s_{\frac{d}{2}+1} & \cdots & s_{T-\frac{d}{2}} \\ \vdots & \ddots & \vdots \\ s_{d+1} & \cdots & s_T \end{pmatrix}, S_F^S \overset{\text{def}}{=} \begin{pmatrix} s_{\frac{d}{2}+2} & \cdots & s_{T-\frac{d}{2}+1} \\ \vdots & \ddots & \vdots \\ s_{d+1} & \cdots & s_T \end{pmatrix} \tag{30}$$

Here, *d* is an even integer indicating the size of a sliding window. Note that corresponding columns of S_H and S_F are "history-future" pairs within sliding windows of size *d*; compared with S_F^+, S_F^S is just missing the first row. We will use the Hankel matrices of both *U* and *Y* in the following steps.

Step 1: oblique projections.

If the true model is an LDS, i.e., everything is linear Gaussian, the expected future observations can be expressed linearly by history observations, history inputs, and future inputs. Formally:

$$\mathbb{E}(Y_F | Y_H, U_H, U_F) = [\beta_{Y_H}\ \beta_{U_H}\ \beta_{U_F}] \begin{bmatrix} Y_H \\ U_H \\ U_F \end{bmatrix} \tag{31}$$

Here, $\Theta = [\beta_{Y_H}\ \beta_{U_H}\ \beta_{U_F}]$ is the linear coefficient that could be solved by:

$$\widehat{\Theta} = \left[\hat{\beta}_{Y_H}\ \hat{\beta}_{U_H}\ \hat{\beta}_{U_F}\right] = Y_F \begin{bmatrix} Y_H \\ U_H \\ U_F \end{bmatrix}^{\dagger} \tag{32}$$

where † denotes the Moore-Penrose pseudo-inverse.

Similar as in Section 5.4.1, we can also solve Θ by further considering a *group lasso* [40] penalty, which is to find the optimal parameters by solving:

$$\min_{\Theta}\left(\left\|Y-\Theta\begin{bmatrix}Y_H\\U_H\\U_F\end{bmatrix}\right\|_2^2+\lambda\sum_{i}\sum_{l=1}^{L}\sqrt{p_l}\left\|\Theta_{\mathrm{i}_l}\right\|_2\right) \tag{33}$$

However, since in a real-time scenario the future input, U_F, is unknown, we want to keep the causality in the training step by partially explaining future observations based on the history. In other words, we care about the best estimation of future observations but just based on the history observations and inputs. Formally,

$$\hat{O}_F \overset{\text{def}}{=} \hat{\Theta}_H\begin{bmatrix}Y_H\\U_H\\0\end{bmatrix}=\left[\hat{\beta}_{Y_H}\ \hat{\beta}_{U_H}\ 0\right]\begin{bmatrix}Y_H\\U_H\\0\end{bmatrix}=Y_F\begin{bmatrix}Y_H\\U_H\\U_F\end{bmatrix}^{\dagger}\begin{bmatrix}Y_H\\U_H\\0\end{bmatrix} \tag{34}$$

where $\hat{O}_F$ is referred to as the oblique projection of Y_F "along" U_F and "onto" $\begin{bmatrix}Y_H\\U_H\end{bmatrix}$.

Geometrically, we are projecting the future observations on the space of the history, but we are missing the future inputs. This is why the projection is called "oblique." In terms of music performance, this oblique projection reveals how much we can predict the next few steps of 2nd piano's performance without observing the future performance of the 1st piano. In this step, we also use the same technique to compute $\hat{O}_F^{+}$ and just throw out its first row to obtain $\hat{O}_F^{S}$.

Step 2: state estimation by singular value decomposition (SVD)

If we know the true parameters of the LDS, the oblique projections and the hidden states would have the following relationship:

$$\hat{O}_F=\mathit{\Gamma}_f Z_f \overset{\text{def}}{=} \begin{bmatrix}C\\CA\\\vdots\\CA^{\frac{d}{2}-1}\end{bmatrix}\left[z_{\frac{d}{2}+1},z_{\frac{d}{2}+2},\dots,z_{T-\frac{d}{2}}\right] \tag{35}$$

$$\hat{O}_F^S=\mathit{\Gamma}_f Z_f^S \overset{\text{def}}{=} \begin{bmatrix}C\\CA\\\vdots\\CA^{\frac{d}{2}-1}\end{bmatrix}\left[z_{\frac{d}{2}+2},z_{\frac{d}{2}+3},\dots,z_{T-\frac{d}{2}+1}\right] \tag{36}$$

Intuitively, the information from the history observations and inputs "concentrate" on the nearest future hidden state and then spread out onto future observations. Therefore, we can gain another perspective of the relationship between the future observations and the history by looking at the relationship between the future observations and the nearest hidden state.

The good news is that we have already computed the oblique projections, which are partially estimated future observations, in the last step. Therefore, if we perform SVD on the oblique projections, i.e.,

$$\hat{O}_F = \mathcal{U}\Lambda\mathcal{V}^T \quad \textbf{(37)}$$

the hidden states can be estimated by:

$$\Gamma_f = \mathcal{U}\Lambda^{\frac{1}{2}} \quad \textbf{(38)}$$

$$\hat{Z}_f = \Gamma_f^{\dagger}\hat{O}_F \quad \textbf{(39)}$$

$$\hat{Z}_f^S = \Gamma_f^{\dagger}\hat{O}_F^S \quad \textbf{(40)}$$

since LDS is defined up to a linear transformation. Moreover, if we delete small numbers in Λ and the corresponding columns in $\mathcal{U}$ and $\mathcal{V}$, we essentially enforce a bottleneck on the graphical model representation, learning compact, low-dimensional states.

Step 3: parameter estimation

Once we have estimated the hidden states, the parameters can be estimated from the following two equations:

$$\hat{Z}_f^S = A\hat{Z}_f + BU_f^S + e_w \quad \textbf{(41)}$$

$$Y_f = C\hat{Z}_f + DU_f + e_v \quad \textbf{(42)}$$

We could also estimate the variance and covariance but do not need these parameters during filtering because we never receive ground truth observations. Here, Y_f and U_f are the 1st rows of Y_F and U_F, i.e., $Y_f = \left[y_{\frac{d}{2}+1}, y_{\frac{d}{2}+2}, \dots, y_{T-\frac{d}{2}}\right]$, $U_f = \left[u_{\frac{d}{2}+1}, u_{\frac{d}{2}+2}, \dots, u_{T-\frac{d}{2}}\right]$. Similarly, U_f^S is the 1st row of U_F^S, i.e., $U_f^S = \left[u_{\frac{d}{2}+2}, u_{\frac{d}{2}+3}, \dots, u_{T-\frac{d}{2}+1}\right]$.

A summary of the spectral learning procedure

In summary, the spectral method does three regressions. The first two are reduced-rank partial regressions, which estimate the hidden states by oblique projections and SVD. The third one estimates the parameters. From the perspective of instrumental regression, the oblique projections can be seen as de-noising the latent states by using past observations, while the SVD adds low-rank constraints.

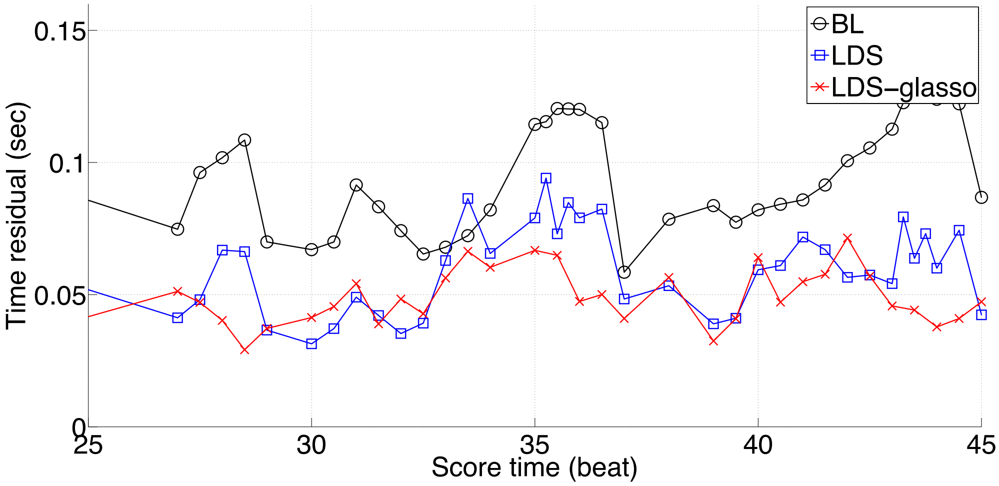
As opposed to maximum likelihood estimation (MLE), the spectral method is a method-of-moments estimator that does not need any random initialization or iterations. Also note that we are making a number of arbitrary choices here (e.g., using equal window sizes for history and future), not attempting to give a full description of how to use spectral methods. (See Van Overschee & De Moor's book [42] for the details and variations of the learning methods.)

5.5.4 Results

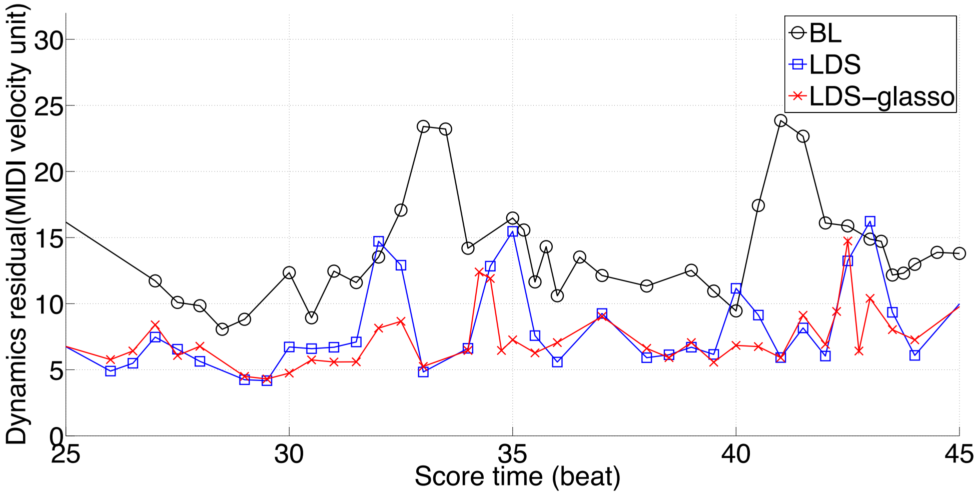
Figure 24 shows the cross-validation result of the LDS, where we see much better result than the baseline trained on only 4 rehearsals. For most pieces, we set the dimension of hidden states n = 5 and the training window size d = 20. Again, (a) and (b) show the detailed results while (d) and (e) show the overall results.

For (a) and (b), the curve with circle markers represents the baseline method, the curve with square markers represents the raw LDS method, and the curve with "x" markers represents the LDS approach with group lasso penalty.

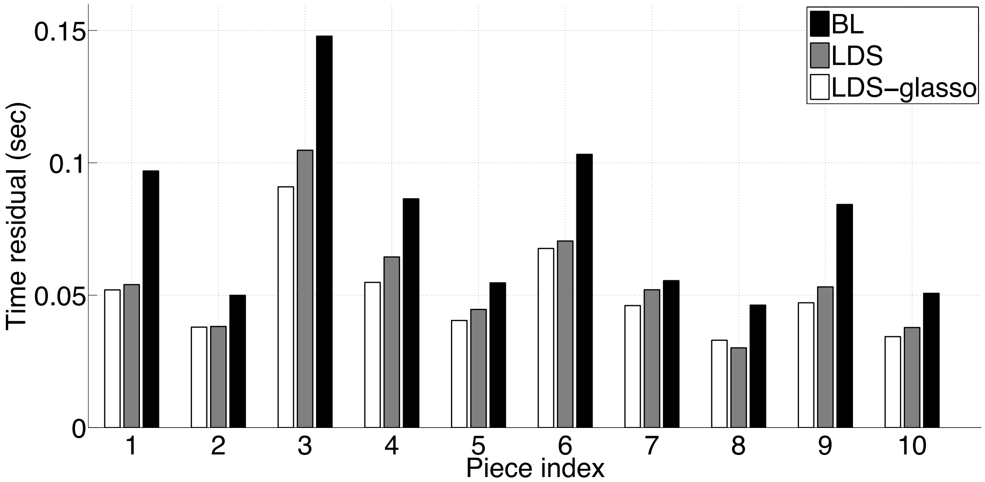
For (d) and (e), the black bars represent the baseline method, the grey bars represent the raw LDS method, and the white bars represent the LDS approach with group lasso penalty. Again, we see that the results are very consistent across different pieces — 4 rehearsals is now enough for all pieces to have a much better prediction than the baseline.



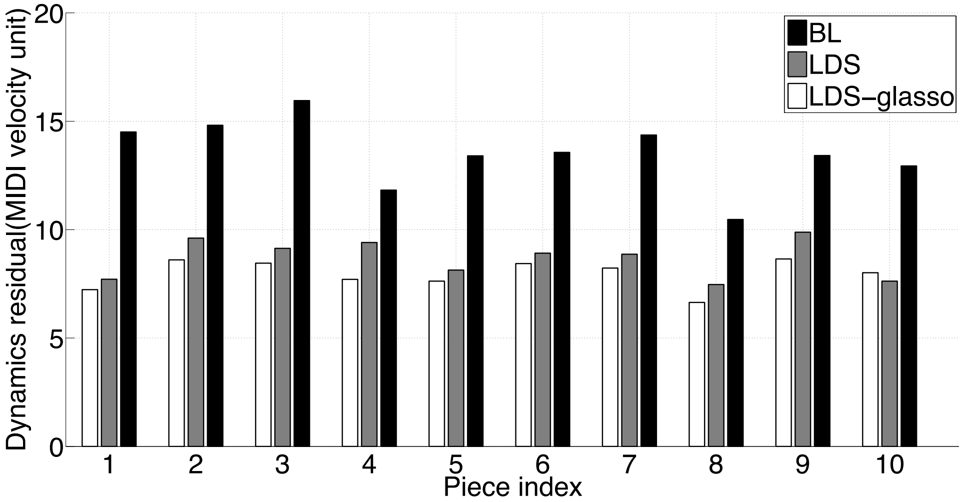
(a) A detailed view of starting time results.



(b) A detailed view of dynamics results.



(c) A global view of starting time results.



(d) A global view of dynamics results.

Figure 24. The cross-validation results of the LDS approach trained on 4 rehearsals. (Smaller is better.)

5.5.5 Discussion

When trained on only 4 rehearsals, the LDS approach improves noticeably compared with the general-feature approach. This indicates that the hidden states layer (introduced in Section 5.5.1) is beneficial to predict the expressive features. Through cross-validation, we see that when the dimensions of the hidden mental states are between 4 and 7, the prediction accuracies for both timing and dynamics are the best. In other words, though the dimensionality of the system input (variable *U* in Section 5.5.1) is much larger (between 60 and 100), the "latent expressive space" (variable *Z* in Section 5.5.1), which controls the expressive performance of the 2nd piano, only contains about 4 to 7 dimensions.

Discovery 7: *There exists a latent expressive space, which explains a significant portion of the musical expression of the 2nd piano. The dimensionality of this latent space is only 4 to 7.*

Another observation is that compared with general-feature approach, the improvement of dynamics prediction is much more obvious than the improvement of timing prediction. Remember that the LDS approach allows information to flow along the time series, adding constrains to each note's musical expression by its preceding notes. Therefore, though the features used for the LDS approach are almost identical to the features used for the general-feature approach, the LDS approach can capture information encoded in a larger scale of music context.

Discovery 8: *Expressive timing is more related to smaller-scale music context compared with expressive dynamics*.

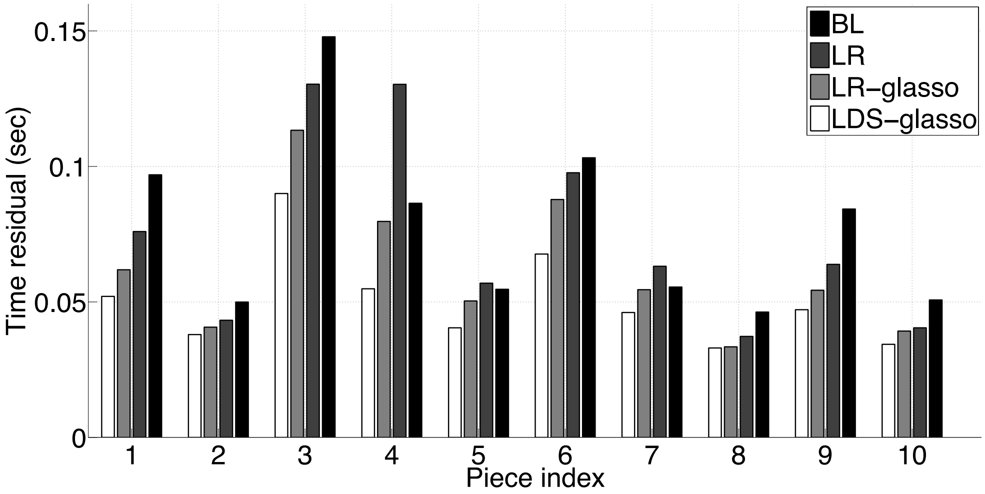
5.6 The overall experimental results

In Section 5.1 through Section 5.5, we have only shown the experimental results for each individual approach. In this section, we show an overall view of the experimental results of different approaches. We first inspect which of the practical approaches in Section 5.6.1, i.e., when trained on only 4 rehearsals, outperforms the baseline. We then introduce alternative measurements from Section 5.6.2 to Section 5.6.4, showing that the results of alternative measurements are consistent with the results measured by absolute residuals. Finally, we inspect how the training set size affects the performance in Section 5.6.5.

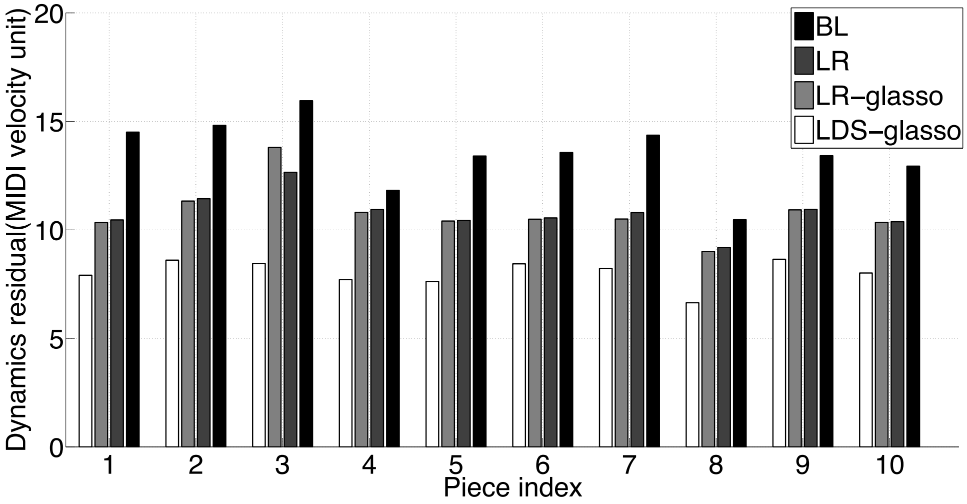
5.6.1 Only 4 rehearsals: a deeper inspection

With 4 rehearsals, which is a practical number for music performance, we start to see better results than the baseline from the general-feature approach (Section 5.4). Figure 25 shows the comparison of the absolute residuals between the baseline and the representative practical models, where subfigure (a) is the timing result and subfigure (b) is the dynamics result.

For both subfigures, the black bars represent the baseline. The dark grey bars represent the linear regression model, which considers various aspects of local music context. The light grey bars add group lasso regularization on top of the linear regression model, throwing out useless feature groups. The white bars represent the LDS model, which further considers the regularization from the musical expression of adjacent notes.



(a) Timing results for all pieces.



(b) Dynamics results for all pieces.

Figure 25. An overall view of different practical models' absolute residuals trained on 4 rehearsals.

We see that the LDS approach performs the best. Compared with the baseline, it can improve the timing prediction by 50 milliseconds and the dynamics (loudness) prediction by 8 MIDI velocity units on average, especially when the baseline is not making good predictions. This is a *very significant* improvement, as listeners can easily perceive asynchronous notes that differ by 30 milliseconds and dynamics differences of 4 MIDI velocity units. The just noticeable difference (JND) for time shifts in otherwise equally-spaced onsets is about 10 milliseconds [77], but this depends on the inter-onset time of the notes and other factors (we are more sensitive when the inter-onset time is smaller, when the context consists of equal inter-onset times, and when the sounds are similar.)

Main result 1: *Trained on only 4 rehearsals, the best machine learning model can shrink the difference between machine prediction and human performance as much as 50 milliseconds (for timing) and 8 MIDI velocity units (for dynamics) compared with the baseline.*

5.6.2 Alternative measurement 1: Average performance difference

Besides the absolute residual between the prediction and human performance, another way to objectively measure the quality of the machine prediction is to inspect the mean difference between the 1st piano (human performance) and the 2nd piano (machine prediction).

Timing: Though the two piano performances are not supposed to be perfectly synchronized, to keep the musicality of a performance, they should not differ too much from each other. Keller et al. [78] showed that the median absolute asynchrony in a piano duet is about 30 to 35 milliseconds (for three pieces chosen for the study). Similarly, we use the *average absolute asynchrony* between the machine prediction and human performance as another measurement. To compute the asynchrony, we only consider the notes that share the same score time in the first and second pianos.

Figure 26 shows that the learning-based models (trained on 4 rehearsals) have smaller average absolute asynchrony compared with the baseline. Here, the *x* axis is the piece index and the *y* axis is the average absolute asynchrony. The black bars represent the baseline, the green bars represent the raw linear regression model, the blue bars represent linear regression with group lasso penalty, the red bars represent the LDS with group lasso penalty, and the white bars represent the ground truths of human performances.

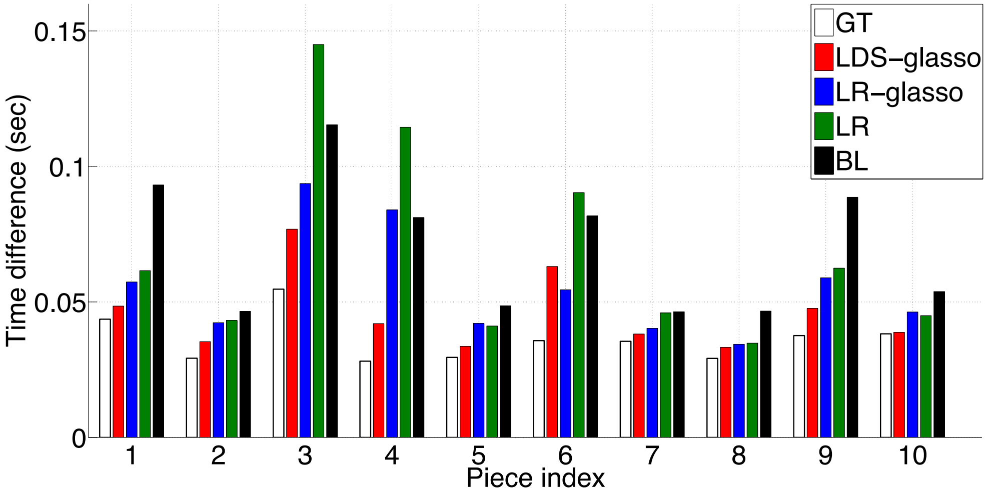


Figure 26. An overall view of average absolute asynchrony between two piano performances (trained on 4 rehearsals).

For most pieces, the average asynchrony result agrees with the result of timing residual result showed in previous sections, even though we are optimizing for timing residual. The learning-based methods shrink the asynchrony dramatically compared with the baseline, especially when the baseline has large asynchronies.

Main result 2: *Trained on 4 rehearsals, the best machine learning model can shrink the absolute asynchrony between the two piano performances as much as 40 milliseconds compared with the baseline.*

It is also important to notice that even the ground truth (human) performances have asynchrony because we are measuring the difference between the two piano performances. The average absolute asynchronies of ground truths are very consistent across different pieces of music and our results agree with the discovery in [78]. (We also inspected the asynchrony without taking the absolute value. Without much surprise, we saw a near zero average timing difference.) We see that human performances still have smaller absolute asynchrony, i.e., tighter timing difference, between the two piano performances compared with all the computational approaches. However, the best learning-based model is very close to the human performance for most pieces.

Main result 3: *In terms of the average absolute asynchrony between the two piano performances, the difference between human performance and the best machine learning model, trained on 4 rehearsals, is only about 10 milliseconds.*

Dynamics: Unlike timing difference, the dynamics of the two piano performances *can* sometimes differ a lot without losing musicality. Therefore, we should not use small dynamics difference as a criterion for a high quality 2nd-piano performance. Nevertheless, we can inspect the average dynamics differences associated with the human performance and different computational models, as shown in Figure 27. Here, different computational models are represented in the same colors and legends as in Figure 26.

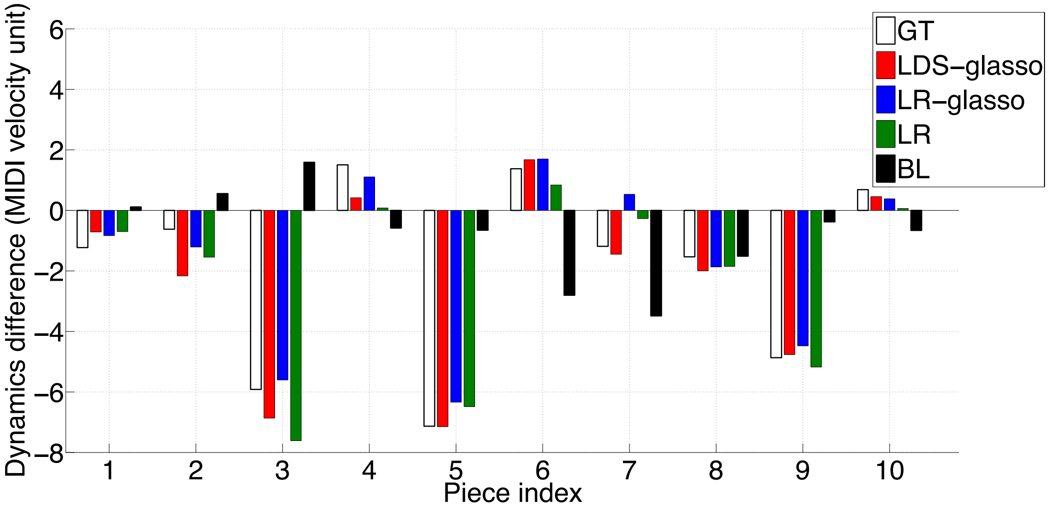


Figure 27. An overall view of average dynamics difference between two piano performances (trained on 4 rehearsals).

We see that the dynamics differences between the two piano performances are very different from their timing differences. On average, the ground truth human performances (represented by the white bars) show negative results. In other words, the

2nd piano performance is *softer* than the 1st piano performance on average. In addition, the dynamics differences of difference pieces are not consistent compared with the timing differences (shown in Figure 26).

Discovery 9: *In terms of the difference between two piano performances, timing difference (asynchrony) is much more consistent across difference pieces of music compared with dynamics difference (loudness balance).*

We also see that the learning-based methods yield much closer results to the ground truths compared with the baseline.

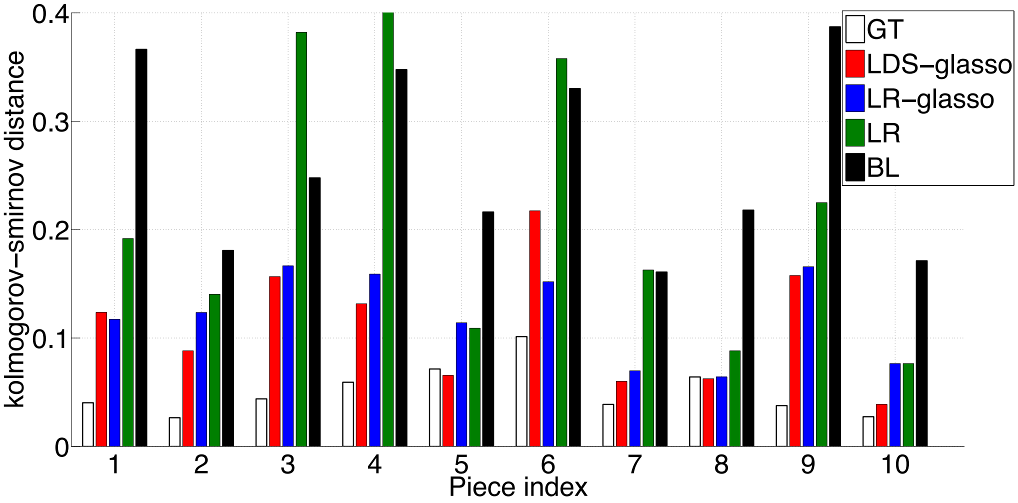
Main result 4: *In terms of the average dynamics difference between the two piano performances, the differences between human performance and the best machine learning model, trained on 4 rehearsals, are under 2 MIDI velocity units for all the pieces.*

5.6.3 Alternative measurement 2: Kolmogorov–Smirnov distance

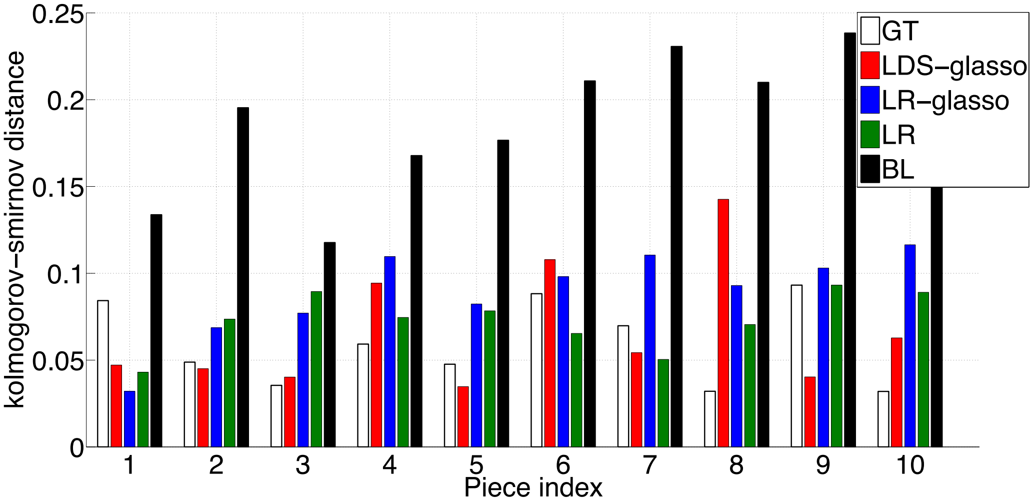
Besides taking the average of the difference between two piano performances, we also inspect another important statistic — the *Kolmogorov–Smirnov* distance [79]. In our case, the Kolmogorov–Smirnov statistic measures the distance between two empirical cumulative distribution functions (CDF) of performance difference by taking the maximum absolute value. Formally,

$$D_n = \sup_x |F_n(x) - F(x)| \tag{43}$$

Here, D_n is the *Kolmogorov–Smirnov* statistic, $F(x)$ is the empirical CDF of the absolute asynchrony/dynamics difference for human performance, and $F_n(x)$ is the empirical CDF of the asynchrony/dynamics difference for a specific computational model. Intuitively, the measurement describes how the asynchrony/dynamics difference associated with machine generation is different from the asynchrony/dynamics difference of human performance. Therefore, smaller numbers yield better results.



(a) Timing distances.



(b) Dynamics distances.

Figure 28. An overall view of the Kolmogorov–Smirnov statistics (trained on 4 rehearsals).

Figure 28 shows the Kolmogorov–Smirnov distances between the ground truth performance difference and the performance differences of computational approaches, where subfigure (a) shows the timing distances and subfigure (b) shows the dynamics distances for different pieces. Here, different computational models are represented in the same colors as in Figure 26. The white bars represent the Kolmogorov–Smirnov distances between human performances by randomly splitting human performances into two halves. Again, we see that the Kolmogorov–Smirnov distances agrees with the results of absolute residuals. For most pieces, the learning-based methods shrink the Kolmogorov–Smirnov distances dramatically compared with the baseline, especially when the baseline has large asynchrony. We also see the distances within human performances are still smaller than any developed computational models.

Main result 5: *In terms of the Kolmogorov–Smirnov distance from the ground truths, the best machine learning model, trained on 4 rehearsals, can shrink the distance as much as 0.35 for timing and 0.2 for dynamics compared with the baseline.*

In addition, the Kolmogorov–Smirnov statistic can be used for the two-sample Kolmogorov–Smirnov test (K-S test). By referring to the notation in equation (**43**), the test determines whether $F(x)$ is significantly different from $F_n(x)$. Our two-sample K-S tests show that for all the pieces of music and all the computational methods, both timing and dynamics Kolmogorov–Smirnov statistics lead to significant difference between $F(x)$ and $F_n(x)$. In other words, though the best machine-learning model shrinks the distance from human performance, their difference is still statistically significant.

5.6.4 Alternative measurement 3: Subjective evaluation

Besides the two *objective* alternative measurements, we invited people to *subjectively* rate our models. To be specific, we designed a double-blind online survey, in which we randomly selected 10 performances from our dataset with one performance per piece of music. During the survey, for each selected piece, subjects first listened to the first piano performance (the melody part) alone, and then listened to *three* duet versions:

- *BL*: The second piano is generated by the baseline model.
- *ML*: The second piano is generated by the best machine-learning algorithm.
- *GT*: The original human duet performance in the dataset.

Note that all the three versions share exactly the same first piano part, which was shown to each subject at first. In addition, since the experiment requires careful listening and a long survey could decrease the quality of answers, each subject only listened to 4 out of the 10 performances by random assignment. The order was also randomized both within a performance (for different duet versions) and across different performances. The total music listening time of each survey is about 17 minutes.

After listening to each duet version, subjects were asked to rate the second piano part in the duet performance on a 5-point Likert scale from 1 (very low) to 5 (very high) according to three criteria:

- *Musicality*: How musical the performance was.
- *Interactivity*: How close the interaction was between the two piano parts.
- *Naturalness*: How natural (human-like) the performance was.

A total of n = 62 subjects (16 female and 36 male) have completed the survey. The aggregated result (as in Figure 29) shows that our best machine-learning model improves the subjective rating significantly compared with the baseline. However, it is still significantly lower than the human performance.

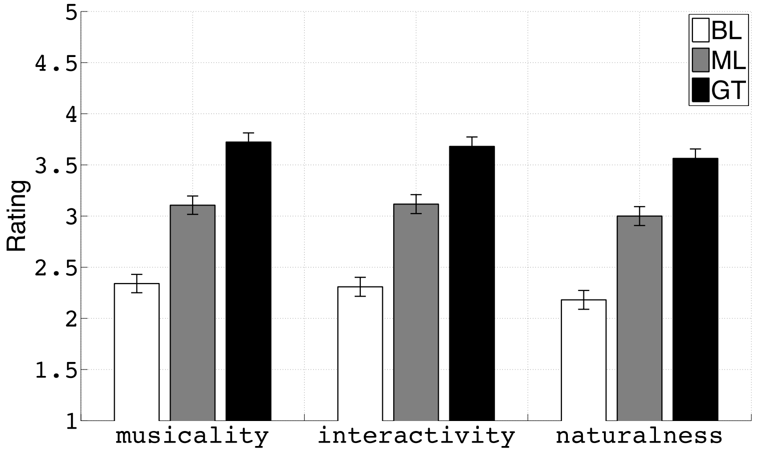


Figure 29. The subjective evaluation results of the duet performance. (Higher is better.)

Here, different colors represent different conditions (versions). The heights of the bars represent the means of the ratings and the error bars represent the MSEs computed via repeated measurement ANOVA. For all three criteria, the p-values are smaller than 0.005.

Main result 6: *Our best model improves the subjective rating by about 0.9 units (in terms of a Likert scale from 1 to 5) compared with the baseline. Compared with ground truth human performance, the rating for our best model is still about 0.5 units lower.*

5.6.5 More rehearsals

We have already seen that the best computational model is the LDS approach with group lasso penalty in Section 5.6.1. We have also seen that this result is consistent under difference measurements from Section 5.6.2 to Section 5.6.4. In this Section, we inspect how much we can gain by adding more rehearsals. To be precise, we test the absolute residuals between machine prediction and the ground truth of human performance under 4, 8 and 16 rehearsals. Since not all the pieces have enough rehearsals, we only show the results for three pieces: *Danny Boy, Ashokan Farewell,* and *Serenade.*

To have a more systematic evaluation, we bring in another computational model (not designed in the thesis) as a comparison — an artificial neural network (ANN). This result was partially contributed by Yun Wang and published in [80]. The purpose is to compare the linear dynamic system with a non-linear model based on the average residuals and the effect of training set size. To be specific, the neural network uses exactly the same feature representation as in the LDS approach. It has a single hidden layer. The hidden layer consists of 10 neurons and uses rectified linear units (ReLUs) to produce non-linearity; the single output neuron is linear. Referring to the notations in Section 5.5, the neural network represents the following relationship between the input feature *U* and the output feature *Y*:

$$Z = f(W_1 U + b_1) \qquad \textbf{(44)}$$

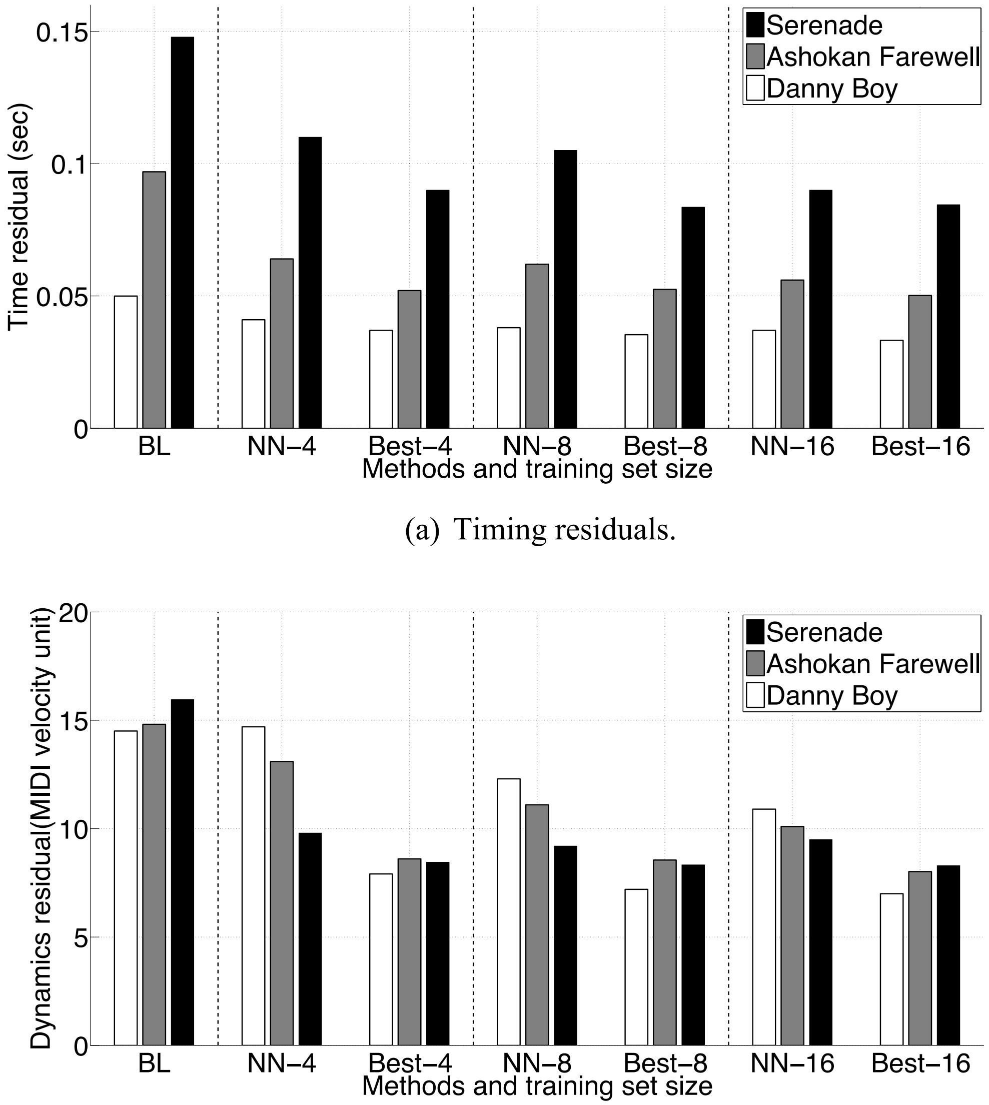
$$Y = W_2 Z + b_2 \qquad \textbf{(45)}$$

where *Z* denotes the activation of the hidden units and

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \qquad \textbf{(46)}$$

The neural network is trained by the minibatch stochastic gradient descent (SGD) algorithm, using the mean absolute error as the cost function. The parameters of the neural network (W_1, b_1, W_2, b_2) are initialized randomly, after which they are tuned with

30 epochs of SGD. Each minibatch consists of one rehearsal. The learning rate decays from 0.1 to 0.05 in an exponential fashion during the training. We report the average absolute and relative errors across five runs with different random initializations on the test set.



(a) Timing residuals.

(b) Dynamics residuals.

Figure 30. A comparison of the cross-validation results between different models with different training set sizes. (Smaller is better.)

Figure 30 shows the comparison of the cross-validation results of absolute residual between the baseline, our best computational model (LDS-glasso approach), and the ANN, trained on 4, 8, and 16 rehearsals. Subfigure (a) shows the timing results, and subfigure (b) shows the dynamics results. Here, for both subfigures, the *x*-axis represents different methods with different training set sizes, the *y*-axis represents the average absolute residual, and different colors represent different pieces. For example, the grey bar above the label "NN-4" denotes the average absolute residual for *Serenade* using the neural network approach with 4 training rehearsals. Results for different training set sizes are separated by dotted lines.

We see that our best model outperforms both the baseline and the ANN approach under all training set sizes. This difference is more obvious with 4 to 8 rehearsals. (From the dynamics result, we see a clear trend that the ANN approach is catching up given more rehearsals.) Though we see improvements of our best model trained on more rehearsals, such improvements are less obvious, especially when a 4 rehearsals training set is producing a low residual.

Main result 7: *Our best model outperforms ANN under all training set sizes that we tried and converges much faster. When trained on 16 rehearsals, the improvement on average is 7 milliseconds (for timing) and 0.5 MIDI velocity unites (for dynamics) compared with 4 rehearsals.*

5.6.6 A summary of evaluation

To summarize the experimental results, our best computational model is LDS with group lasso penalty. It shrinks the absolute difference between machine prediction and human performance as much as 50 milliseconds (for timing) and 8 MIDI velocity units (for dynamics) compared with the baseline, and such results are robust against different objective and subjective measurements. When trained on 16 rehearsals, the improvement on average is 7 milliseconds (for timing) and 0.5 MIDI velocity unites (for dynamics) compared with 4 rehearsals.

Chapter 6

Training Strategies

In the previous chapter, we have presented different computational models and shown that the models outperform the baseline based on different measurements. The focus was to explore better models and learning algorithms that capture the interaction of musical expression in piano duets. In this chapter, we present how different training strategies, i.e., selections of training data sets, affect the experimental results. The focus of this chapter is to further explore the generality of the best learning-based model as well as to gain smarter data selection strategies under realistic constraint of data collection.

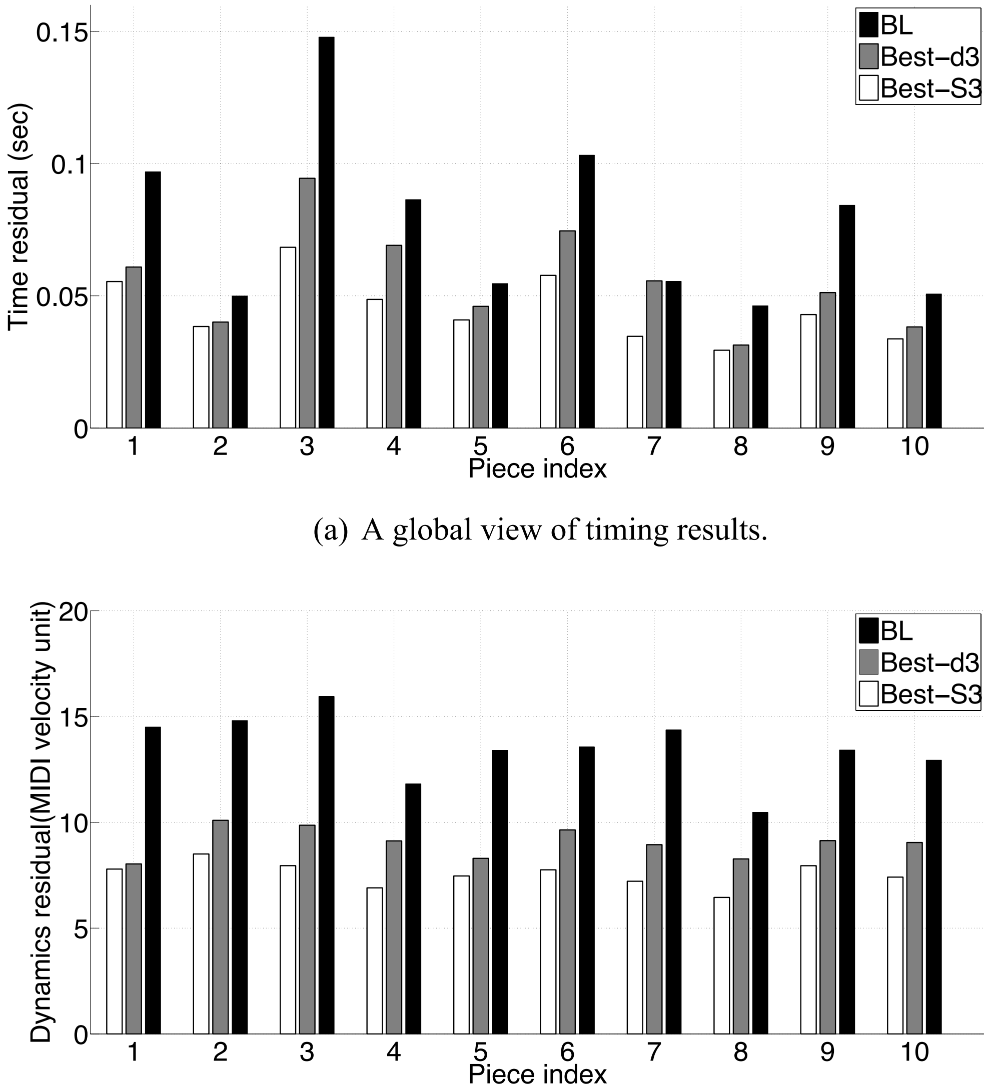
We first present the benefits of training the model on the rehearsals performed by the *same* musicians in Section 6.1. In Section 6.2, we show that without the rehearsals of the same musicians, we can select a relevant subset of rehearsals from *other* musicians' rehearsals and achieve better predictions compared with randomly selecting the rehearsals. In Section 6.3, we show that even without the rehearsals of the same piece of music, we can still outperform the baseline by learning from rehearsals of *other* pieces.

6.1 Same-performer approach

Remember that in the previous chapter, we excluded the rehearsals performed by the *same* pair of performers to avoid overfitting from a machine-learning perspective. From the perspective of a training strategy, all the models in the last chapter belong to a *different-performer approach*. However, the concept of overfitting is very relative in a realistic music scenario since musicians normally rehearse and perform with the same group of people. In this section, we explore whether there is any advantage by a *same-performer approach*. In other words, we learn only from the rehearsals performed by the *same* pair of musicians.

6.1.1 Results

Since each pair of musicians only performs 4 times for most pieces of music, we set the training set size to be 3 to compare the same-performer approach with the different-performer approach. Similar to the last chapter, when the maximum possible training set size is larger than 3, we randomly sample 3 rehearsals from the training set.



(a) A global view of timing results.

(b) A global view of dynamics results.

Figure 31. The cross-validation results of the same-performer approach trained on 3 rehearsals. (Smaller is better.)

Figure 31 shows a comparison between the same-performer approach and the different-performer approach trained on 3 rehearsals, where subfigure (a) shows the timing result and subfigure (b) shows the dynamics result. For both subfigures, the *x*-axis represents the piece index and the *y*-axis represents the absolute difference between machine prediction and human performance (smaller is better). The black bars represent the baseline, the grey bars represent the different-performer approach of our best machine-learning model, and the white bars represent the same-performer approach. We see that for all the pieces, the same-performer approach outperforms the different-performer approach.

Main result 8: *For the best machine learning model trained on 3 rehearsals, the same-performer approach can further shrink the difference between machine prediction and human performance as much as 20 milliseconds (for timing) and 2 MIDI velocity units (for dynamics) compared with the different-performer approach.*

6.1.2 Discussion

The improvement with the same-performer approach indicates that it is a better strategy for musicians to collect and use their own rehearsals if they have this freedom. The improvement also indicates that the musical expression of the same pair of performers in piano duets is *consistent* and this consistency benefits the prediction accuracy.

6.2 Consistent-performance approach

In a realistic scenario, musicians may not have the freedom to collect their own rehearsals with collaborators. They may also have limited access to the rehearsals of other musicians due to privacy or ownership issues. Hence, one question naturally arises: besides using performer identities, are there any other ways to find consistent performances? If the answer is yes, even with only the rehearsals of *other* performers, musicians will still be able to select a relatively consistent subset (to train the model) and achieve better performances compared with randomly selecting the rehearsals.

In this section, we first introduce a measurement to reveal the consistency between performances in Section 6.2.1. Then, we show two approaches to select consistent performances from other musicians' rehearsals in Section 6.2.2. We show the experimental results in Section 6.2.3 and hold the discussion in Section 6.2.4.

6.2.1 A consistency measurement for musical expression

If we need the computer to perform with a given 1st piano performer, we should look for performances where the 1st piano parts are played with an expression that is typical (consistent) of that 1st piano performer. To measure the consistency between different 1st piano performances, we slightly modify the baseline algorithm introduced in Section 5.1.

For timing, we use one 1st piano performance (as *reference* time) to predict another 1st piano performance (as *target* time) using linear regression based on several recently observed notes. Then, the mean absolute difference between the estimated target and the truth target can be understood as a *timing difference* to reveal the consistency of expressive timing. (Note that the timing difference is asymmetric, but in practice flipping reference and target timings leads to very similar results.)

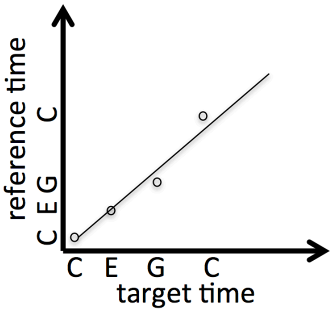


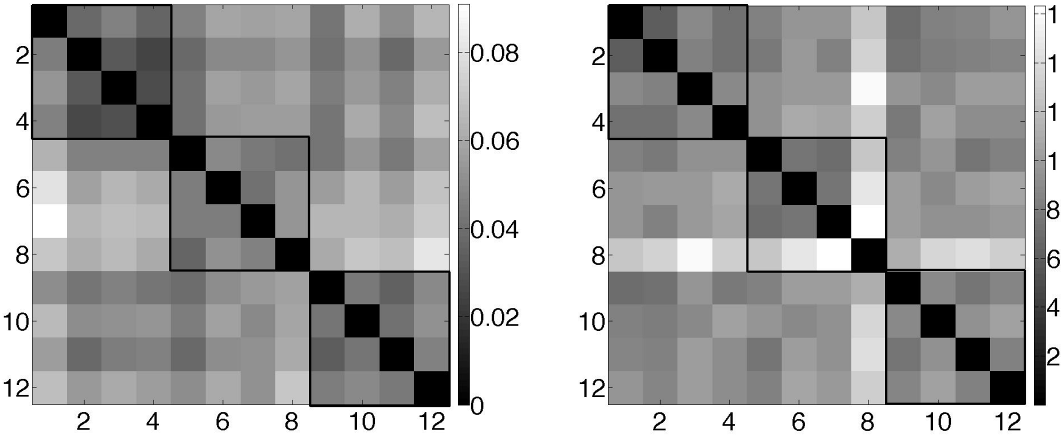
Figure 32. An illustration of consistency measurement for performance timing.

Figure 32 shows an example where the *x*-axis is the target timing and the *y*-axis is the reference timing. The line represents a linear regression that estimates the next note's target time. Clearly, if the target time were identical to the reference time or simply a "stretched" reference, the distance between the two performances would be zero.

For dynamics, we take a first-order difference for both 1st piano performances and use their mean absolute difference as the *dynamic distance*. Formally, let the dynamics of one 1st piano be $d = [d_1, d_2, \ldots, d_i, \ldots]$ and let the dynamics of another 1st piano be $d' = [d_1', d_2', \ldots, d_i', \ldots]$. Then, the dynamics distance between the two performances is:

$$\mathrm{Dist_d} = \frac{1}{n}\sum_{i=2}^{n} |(d_i - d_{i-1}) - (d'_i - d'_{i-1})| \tag{47}$$

Figure 33 shows an example of the pair-wise distance matrix between different performances, where subfigure (a) shows the timing distance and subfigure (b) shows the dynamics distance. For both subfigures, the element at *i*th row and *j*th column represents the distance between the i^{th} and j^{th} performance (using j^{th} performance as the reference). The elements in black rectangles represent the distances between the performances of the same pairs of musicians. Clearly, we see that the distances for the same pairs of musicians are smaller compared with the distances for different pairs of musicians on average. The result indicates that the designed consistency measurement is effective.



(a) Pair-wised timing distances.

(b) Pair-wise dynamics distances.

Figure 33. An illustration of the pair-wise distances between difference performances.

6.2.2 Selection of consistent rehearsals

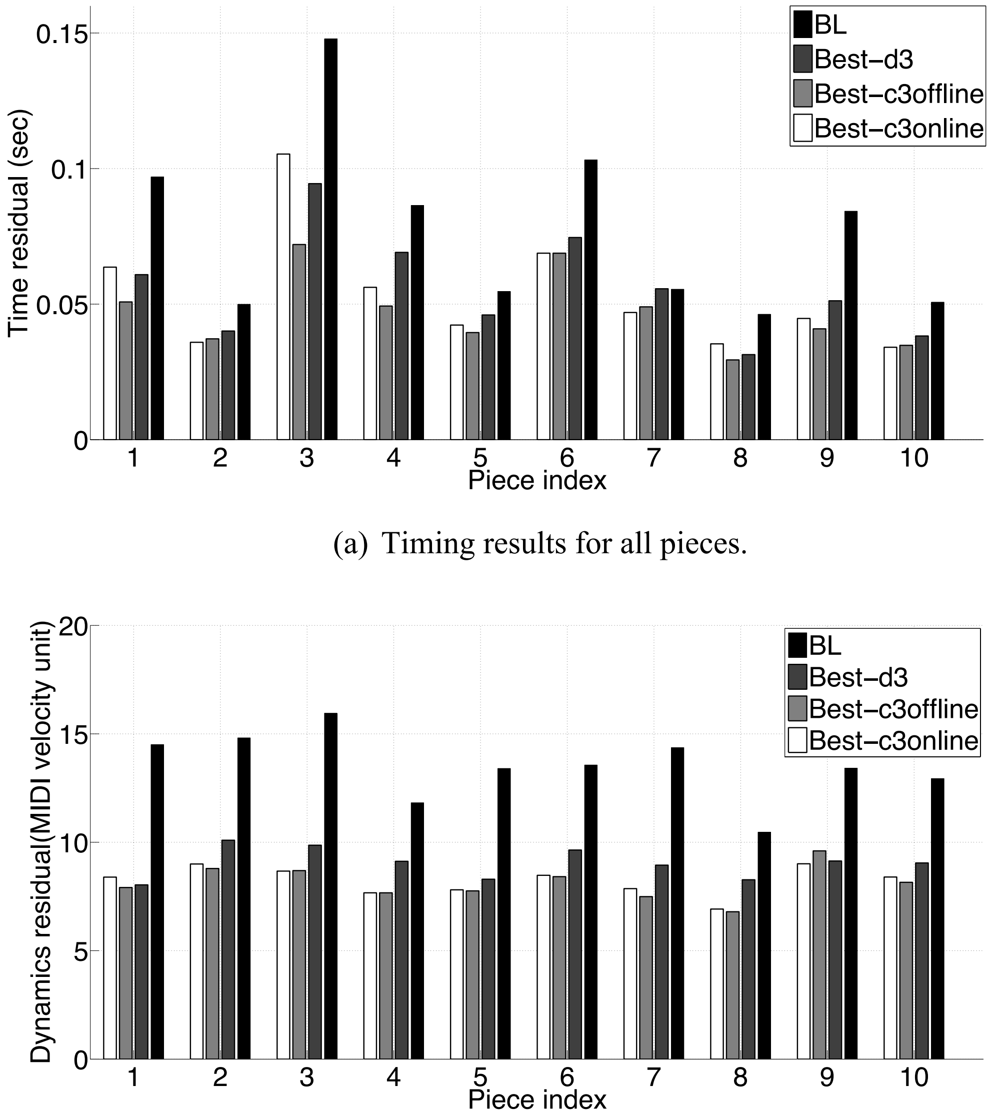
The consistency measurement introduced in the last section allows a musician to select a relevant subset from the rehearsals of *different* musicians using only his or her solo performances. In this section, we present two strategies to select the consistent rehearsals: offline approach and online approach.

Offline approach: Given a musician and a piece of music, we first compute a median rehearsal (introduced in Section 5.1.3) of his/her performances of the piece. This median rehearsal can be seen as an estimation of the performance on stage, based on which a consistent subset of rehearsals is selected.

Online approach: Rather than using a median performance as the estimation, the online approach uses a part (the first half) of the performance in real time to select a consistent subset of rehearsals.

6.2.3 Results

Figure 34 shows the cross-validation results of the consistent-performance approach trained on 3 rehearsals. We see better results than the different-performer approach, i.e., randomly sampling the rehearsals.



(a) Timing results for all pieces.

(b) Dynamics results for all pieces.

Figure 34. The cross-validation results of the consistent-performance approach trained on 3 rehearsals. (Smaller is better.)

Here, subfigure (a) shows the timing result and subfigure (b) shows the dynamics result. The black bars represent the baseline. The dark grey bars represent the different-performer approach (of the best machine-learning model) trained on randomly selected rehearsals. The light grey bars represent the consistent-performance approach where the rehearsals are selected offline. The white bars represent the consistent-performance approach where the rehearsals are selected online based on the first half of the performance.

Main result 9: *For the best machine learning model trained on 3 rehearsals, the consistent-performance approach can further shrink the difference between machine prediction and human performance as much as 15 milliseconds (for timing) and 1.5 MIDI velocity units (for dynamics) compared with the different-performer approach.*

6.2.4 Discussion

The improvement with the consistent-performance approach indicates that it is a better strategy for musicians to collect consistent rehearsals if they have to resort to others' rehearsals but have limited access. We also see that for most pieces, the offline approach is better than the online approach. This result indicates that to collect consistent rehearsals, musicians should at least record their own solo performances before hand.

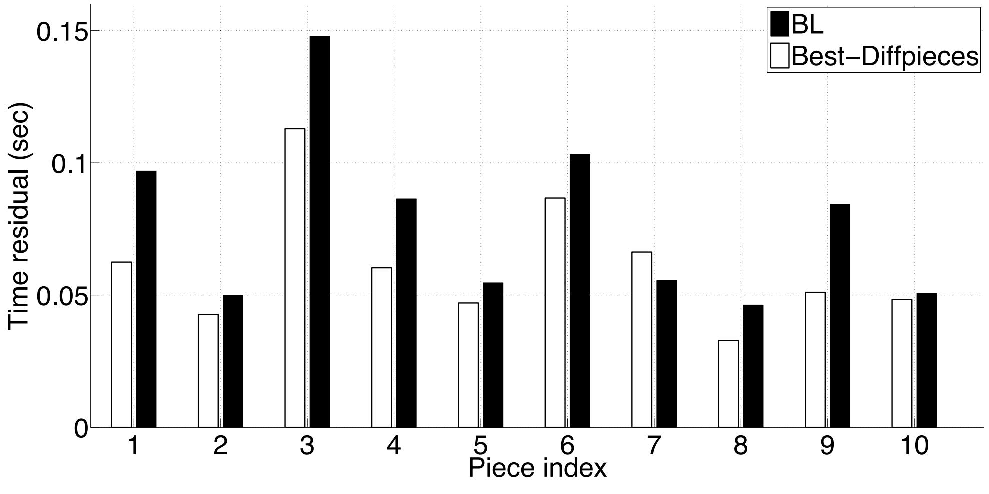
Nevertheless, when compared with the different-performer approach trained on 8 rehearsals (all possible rehearsals for most pieces), the advantage of the consistent-performance approach disappears. (This is not shown in Figure 34.) This result indicates that musicians should still use all the rehearsals if they have the full access. In other words, given the choice of using just a few consistent rehearsals versus many rehearsals, the better approach is to train on many rehearsals. Evidently, using many rehearsals allows the system to learn how to perform under many conditions. Thus the performance with a particular player is not degraded, and the learned model is more robust because it does not assume any particular performance style.

6.3 Cross-piece approach

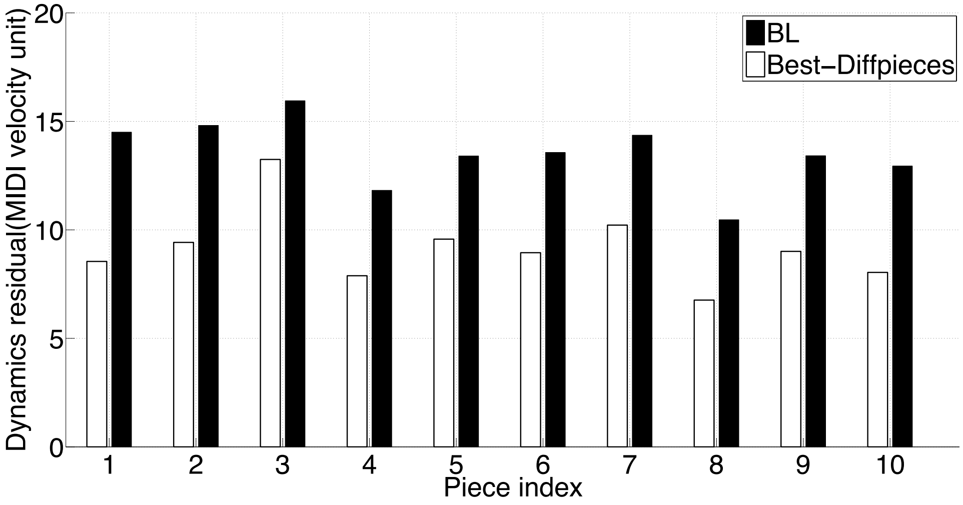
So far, all the approaches (either trained on the same or different performers) are all based on the *same* piece of music. We refer to these approaches as the same-piece approach. In a realistic scenario, musicians may not be able to collect any rehearsals of the piece they want to perform. In this section, we explore whether we can still outperform the baseline by a *cross-piece approach*. In other words, we only learn from the rehearsals of *other* pieces of music.

6.3.1 Results

Figure 35 shows a comparison between the baseline and cross-piece approach trained on all the rehearsals, where we see that for both timing and dynamics, the cross-piece outperforms the baseline for most pieces of music, though not as pronounced when trained on the same pieces. Here, subfigure (a) shows the timing result and subfigure (b) shows the dynamics result. The black bars represent the baseline and white bars represent the cross-piece approach of the best machine learning model trained on all available rehearsals.



(a) Timing results for all pieces.



(b) Dynamics results for all pieces

Figure 35. The cross-validation results of the cross-piece approach trained on all rehearsals. (Smaller is better.)

Main result 10: *For the best machine learning model, the cross-piece approach can shrink the difference between machine prediction and human performance as much as 15 milliseconds (for timing) and 5 MIDI velocity units (for dynamics) compared with the baseline.*

6.3.2 Discussion

The improvement with the cross-piece approach indicates that the expressive musical interaction in a piano duet follows universal patterns. In addition, our best model is capable of generalizing what it has learned and applying it in new situations. However, this approach requires a large number of rehearsals (about 150 to 170 in our dataset) from other pieces of music. Also, the improvement with the cross-piece approach is less pronounced compared with the same-piece approach, which means that musicians should still use the rehearsals of the same piece, if possible.

6.4 A summary of different training strategies

In this chapter, we have presented three different training strategies under realistic constraints of data collection. To be specific, if musicians do not have rehearsals of the same piece, they should use the cross-piece approach. If musicians have limited access to the rehearsals of the same piece and do not have the freedom to collect their own rehearsals, they should use the consistent-performance approach. If musicians have the freedom to collect their own rehearsals of the same piece, they should use the same-performer approach. (We did not explore the "same performer but different piece" case because the corresponding dataset size is not big enough for the learning task.)

All these approaches outperform the baseline on average. The cross-piece approach does not perform as well as the same-piece approach, and the same-piece approach does not perform as well as the consistent-performance approach and same-performer approach. Thus, we conclude that there are multiple factors that determine expressive performance. These factors include general musicianship and universal performance rules (illustrated by cross-piece training), composition-specific structures (illustrated by same-piece training), and performer preferences and performance styles (illustrated by same-performer and consistent-performance training).

Chapter 7

Basic Improvisation Techniques

So far, our study of expressive and collaborative performance has focused on the two most fundamental aspects of musical expression: timing and dynamics. In other words, performances still strictly follow the pitches and rhythms specified in the score. This chapter addresses a new level of musical expression where musicians have the freedom to improvise, that is, insert, delete and modify pitches and rhythms. We aim to extend the computational models developed in previous chapters and apply them to learn basic improvisation techniques in collaborative performance.

Studies of computer-generated improvisation can trace back to the early 1980s. Early systems [81]-[83] incorporated compositional knowledge to created rule-based improvisation. While statistical models were applied to music composition as early as the 1950's [84], work focused on learning-based improvisation [85][86] started to appear in 2000. While most studies consider a more constraint-free jazz scenario, our study considers the improvisation in a folk/classical music scenario performed collaboratively by two musicians. The music to be performed consists of a melody and a chord progression (harmony). In this deliberately constrained scenario, the melody is to be expressed clearly, but it may be altered and ornamented. This differs from a traditional jazz improvisation where a soloist constructs a new melody, usually constrained only by given harmonies. In musical terms, we want to model the situation where a notated melody is marked "*ad lib.*" as opposed to a passage of chord symbols marked "*solo.*" A melody that guides the performance simplifies the learning task and makes the evaluation procedure more repeatable. The second part is simply a chord progression (a lead sheet), which is the typical input for a jazz rhythm section (the players who are not "soloing"). The second player, which we will implement computationally, is free to construct pitches and rhythms according to these chords, supporting the first (human) player who improvises around the melody.

It is important to note that the focus of this chapter is not the *performance* properties of individual notes (such as timing and dynamics) but the *score* properties of improvised collaborative performance. Normally, improvisors play very intuitively, imagining and producing a performance, which might later be transcribed into notation.

In our model, we do the opposite, having our system generates a symbolic score where pitch and rhythm are quantized. To gain training examples of improvised scores, we collected a new piano duet dataset, which contains multiple improvised performances of each piece. Our general solution is to develop a measure-specific model, which computes the correlation between various aspects of first piano performance and the score of second piano performance measure-by-measure. (This can be seen as an extension of the note-specific approach introduced in Section 5.2) Based on the model, an artificial performer constructs an improvised part based on a lead sheet, in concert with an embellished human melody performance.

7.1 Data collection

To learn the expressive interaction with improvisation techniques, we collected a new dataset. It contains two songs: *Sally Garden* and *Spartacus Love Theme*, each performed 15 times by the same pair of musicians. The recording setting and procedure are the same as described in Section 3.1.

An overview of the dataset can be seen in Table 3, where each row corresponds to a piece of music. The first two columns represent piece index and name. The 3rd to 5th columns represent the number of chords (each chord covers a measure on the lead sheet), the average number of embellished notes in the first piano performance, and the average performance length.

Table 3. An overview of the improvised piano duet performance dataset.

index	name	#chord	#avg. emb. note	avg. length
1	Sally Garden	36	27	1’09’’
2	Spartacus Love Theme	20	12	53’’

7.2 Data preprocessing

Improvisation techniques present new challenges for data preprocessing: performances no longer strictly follow the defined note sequences, so it is more difficult to align performances with the corresponding scores. To address this problem, for the first piano part (the melody), we manually aligned the performances with the corresponding scores since we only have 30 performances in total and each of them is very short. For the second piano part, since the purpose is to learn and generate the scores, we want to transcribe the score of each performance before extracting features or learning patterns from it. Specially, since our performances were recorded by electronic pianos with MIDI outputs, we know the ground truth pitches of the score and only need to transcribe the rhythm (i.e., which beat each note aligns to).

We adopted a modification of the baseline algorithm introduced in Section 5.1 for the rhythm transcription of the second piano part. The algorithm contains three steps: *score-time calculation*, *half-beat quantization*, and *quarter-beat refinement*. In the first step, we compute raw score timings of the second piano notes using the local tempi of the aligned first piano part within 2 beats as the guidance. Figure 36 shows an example, where the performance time of the target note is *x* and its score time is computed as *y*. In this case, the neighboring context is from 7th to 11th beat, the "+" signs represent the onsets of the first piano notes within 2 beats of the target note, and the dotted line is the tempo map computed via linear regression.

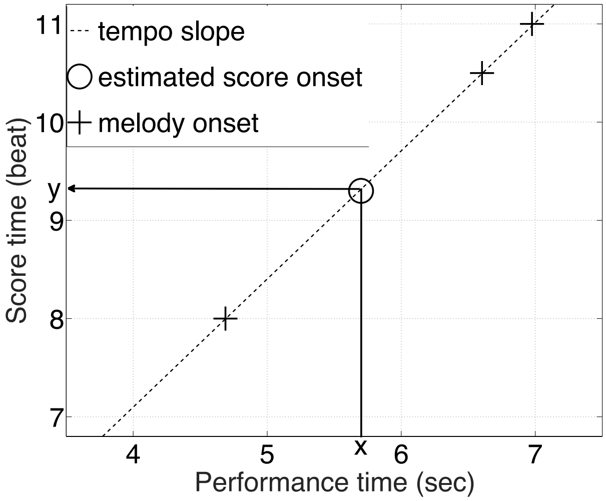


Figure 36. An illustration of score-time calculation for rhythm transcription.

In the second step, we quantize the raw score timings computed in the first step by rounding them to the nearest half beats. For example, in Figure 36, *y* is equal to 9.3 and it will round up to 9.5. In the final step, we re-quantize the notes to ¼ beat if two adjacent notes were quantized to the same half beat in the second step and their raw score time is within the range of ¼ beat ± *error*. In practice, we set the *error* to be 0.07 beat. For the example in Figure 36, if the next note's raw score time is 9.6, the two notes will be quantized to 9.5 in the second step but re-quantized to 9.25 and 9.5, respectively, in the final step. The rationale of the quantization rules is that for our dataset, most notes align to half-beat and the finest subdivision is ¼ beat.

7.3 Feature representations

Similar to the learning task for expressive timing and dynamics, we design input and output features to serve as an intermediate layer between transcribed data (presented in the last section) and the computational model (to be presented in the next section). The input features are designed to represent the score and the 1st piano performance, while the output features are designed to represent the transcribed score of the 2nd piano. However, unlike previous chapters, the *unit* for learning improvisation is a *measure* rather than a note/compound event. The reason is that an improvisation choice, especially the choice of improvised rhythm, of a measure is more of an organic whole than independent decisions on each note, chord, or beat.

7.3.1 Input features

The input features reveal various aspects of the first piano performance that affect the score of the second piano. Remember that the first piano part follows a pre-defined monophonic melody and performers can add embellishments. Formally, we use $x = [x_1, x_2, \ldots, x_i, \ldots]$ to denote the input feature sequence with *i* being the measure index. To be specific, x_i includes the following components:

Tempo Context: The tempo of the previous measure, which is computed by:

$$\mathrm{TempoContext}_i \stackrel{\mathrm{def}}{=} \frac{p_{i-1}^{\mathrm{last}} - p_{i-1}^{\mathrm{first}}}{s_{i-1}^{\mathrm{last}} - s_{i-1}^{\mathrm{first}}} \tag{48}$$

where p_{i-1}^{first}(or s_{i-1}^{first}) and p_{i-1}^{last} (or s_{i-1}^{first}) represent the performance time (or score time) of the first and last note in the previous measure, respectively.

Embellishment Complexity Context: A measurement of how many embellished notes are added to the melody in the previous measure. Formally,

$$\text{EmbellishmentComplexityContext}_i \overset{\text{def}}{=} \log\left(\frac{\#P_{i-1} - \#S_{i-1} + 1}{\#S_{i-1} + 1}\right) \quad \textbf{(49)}$$

where $\#S_{i-1}$ represents the number of notes defined in the score and $\#P_{i-1}$ represents the number of actual performed notes.

Onset Density Context: The onset density of the second piano part in the previous measure, which is defined as the number of score onsets. Note that one chord just count as one onset. Formally:

$$\text{OnsetDensityContext}_i \overset{\text{def}}{=} \#\ \text{Onset}_{i-1} \quad \textbf{(50)}$$

Chord Thickness Context: The chord thickness in the previous measure, which is defined as the average number of notes in each chord. Formally:

$$\text{ChordThicknessContext}_i \overset{\text{def}}{=} \frac{\#\ \text{Note}_{i-1}}{\text{OnsetDensityContext}_i} \quad \textbf{(51)}$$

where $\#\ \text{Note}_{i-1}$ represents the total number of notes in the previous measure.

7.3.2 Output features

For each measure, we focus on the prediction of its *onset density* and *chord thickness.* Formally, we use $y = [y_1, y_2, \ldots, y_i, \ldots]$ to denote the output feature sequence with i being the measure index. By referring to the notations in Section 7.3.1, y_i includes the following two components:

$$\text{OnsetDensity}_i \overset{\text{def}}{=} \#\ \text{Onset}_i \quad \textbf{(52)}$$

$$\text{ChordThickness}_i \overset{\text{def}}{=} \frac{\#\ \text{Note}_i}{\#\ \text{Onset}_i} \quad \textbf{(53)}$$

To map these two features into an actual score, we use *nearest-neighbor search* treating onset density as the primary criteria and chord thickness as the secondary criteria. Given a predicted feature vector, we first search the training examples (score of the same measure for other performances) and select the example(s) whose onset density is/are closest the predicted onset density. If multiple candidate training examples are selected, we then choose the candidate whose chord thickness is closest to the predicted chord thickness. If there are still multiple candidates left, we randomly choose one from them.

7.4 The computational model

We developed a measure-specific approach, which trains a different model for every measure. Intuitively, this approach assumes that the improvisation decision on each measure is linearly correlated to performance tempo, melody embellishments, and the rhythm of the previous measure. Formally, if we use $x = [x_1, x_2,\ldots, x_i,\ldots]$ and $y = [y_1, y_2,\ldots, y_i,\ldots]$ to denote the input and output feature sequences with i being the measure index, the model is:

$$y_i = \beta_0^i + \beta^i x_i \tag{54}$$

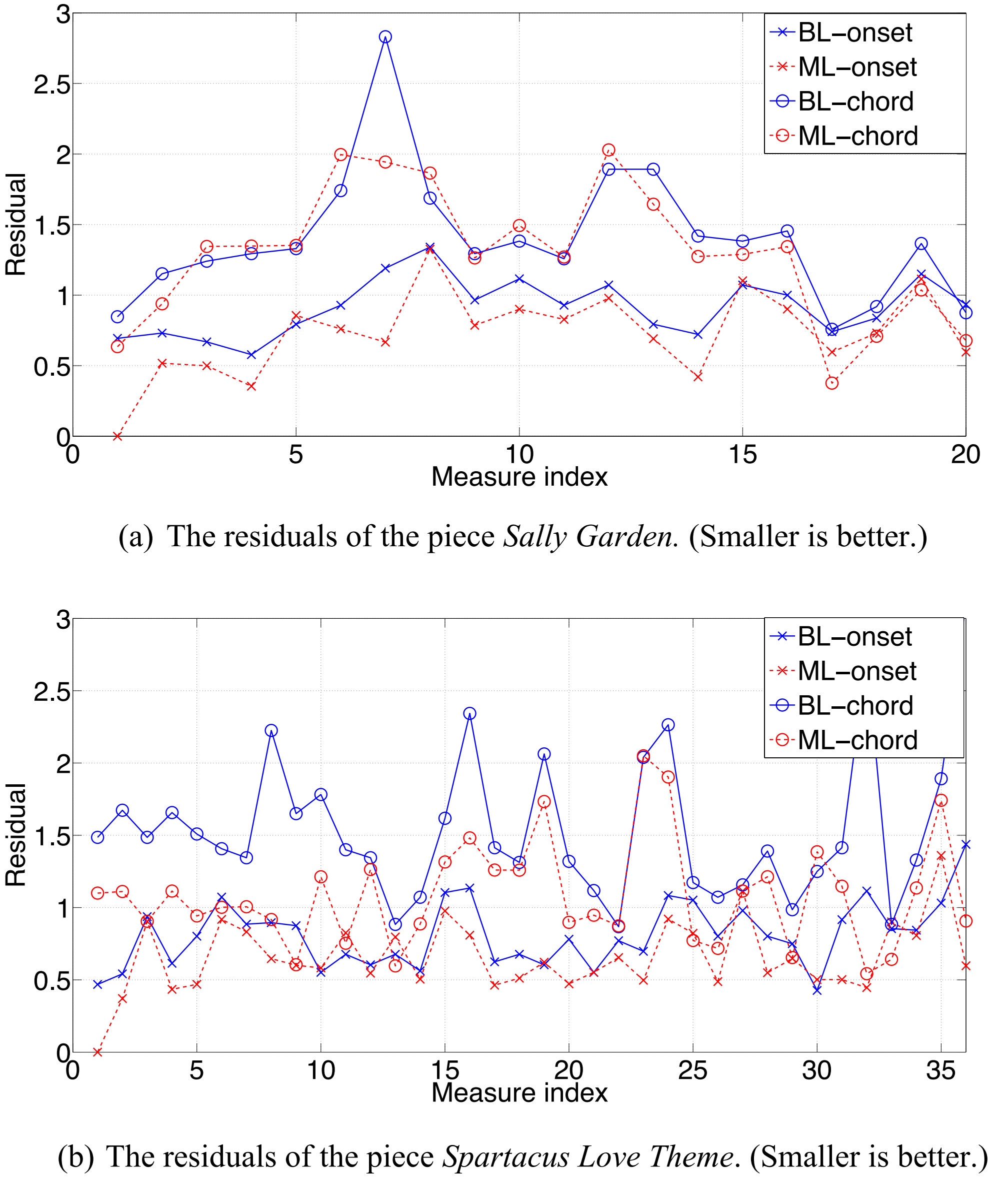
The measure-specific approach is able to model the improvisation techniques even if it does not consider many of the compositional constraints. (For example, what the proper pitches are given a chord, and what the proper choices of rhythm are given the relative position of a measure in the corresponding phrase.) This is because we train a tailored model for each measure and most of these constraints have already been encoded in the training examples. Therefore, when we decode (generate) the performance using nearest-neighbor search on training performances, the final output performance will also meet the compositional constraints.

7.5 Experimental results

We adopted the *mean* of training samples as our baseline prediction and conducted both objective and subjective evaluations. For objective evaluation, we measured the absolute difference between predicted output features and the ground truth output features. For subjective evaluation, we designed a survey and invited people to rate the synthetic performances generated by the designed models.

7.5.1 Objective evaluation

Figure 37 shows the leave-one-out cross-validation results of the measure-specific approach, where we see that it outperforms the baseline. Here, subfigure (a) shows the result for the first piece and subfigure (b) shows the result of the second piece. For both subfigures, the curves with “x” markers are the results for onset density (the primary feature) and the curves with circle markers are the results for chord thickness (the secondary feature). The solid curves represent residuals of the baseline approach (sample means) and the dotted curves represent residuals of the measure-specific approach. Therefore, small numbers mean better results.



(a) The residuals of the piece *Sally Garden*. (Smaller is better.)

(b) The residuals of the piece *Spartacus Love Theme*. (Smaller is better.)

Figure 37. The objective evaluation results.

Main result 11: *On average, the measure-specific model can shrink the difference between machine prediction and human performance as much as 0.2 units (for onset density) and 0.4 units (for chord thickness) compared with the baseline.*

7.5.2 Subjective evaluation

Similar to the subjective evaluation conducted in Section 5.6.4, we designed a survey to evaluate piano duets with basic improvisation techniques according to three criteria: *musicality*, *interactivity*, and *naturalness* on a 5-point Likert scale from 1 (very low) to 5 (very high). We randomly selected four performances from the dataset with two performances per piece and presented them in a random order in the survey. For each performance, we compared the ratings of *three* duet versions:

- *BL*: The score of the second piano is generated by the baseline mean estimation.
- *ML*: The score of the second piano is generated by the measure-specific approach.
- *QT:* The score of the second piano is the quantized original (ground truth) human performance.

The three versions share exactly the same first piano part and their differences lie in the second piano part. As our focus is the evaluation of improvisation of pitch and rhythm, the timing and dynamics of all the synthetic versions are generated using the automatic accompaniment approach introduced in Section 5.1.

A total of n = 42 subjects (13 female and 29 male) have completed the survey. The aggregated result (as in Figure 38) shows that the measure-specific model improves the subjective rating significantly (with the p-values less than 0.05) compared with the baseline for all three criteria. Surprisingly, our method even generates better results than using the score transcribed from original human performances, though the differences are not significant (with the p-values larger than 0.05). Note that this result does not indicate the measure-specific model is better than the *original* human performance because the timing and dynamics parameters are still computed by an automatic accompaniment algorithm for the “GT” version and there are unavoidable quantization errors during the rhythm transcription.

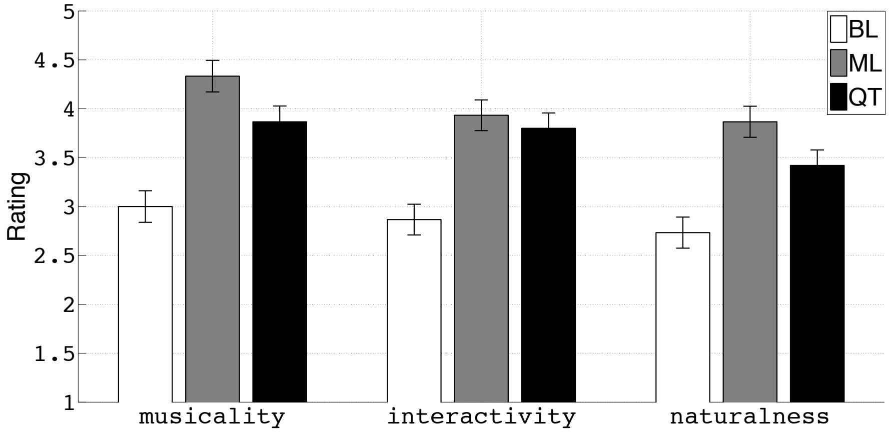


Figure 38. The subjective evaluation results of the duet performances with improvisation techniques. (Higher is better.)

Main result 12: *The measure-specific model improves the subjective rating by about 1 unit (in terms of a Likert scale from 1 to 5) compared with the baseline.*

7.6 Conclusion

In conclusion, we extended the computational models developed in previous chapters and applied them to improvisational aspects of musical expression. The experimental results show that the developed measures-specific approach is able to generate more musical, interactive, and natural collaborative performance than the baseline mean estimation. As the most recent effort of this thesis, our study of improvisation techniques has not yet considered general improvisation rules that apply to different measures or even different pieces of music, complex music structures, or any performer preferences and styles. We leave the study of these aspects of improvisation to the future work.

Previous work on machine learning and improvisation has largely focused on modeling style and conventions as if collaboration between performers is the indirect result of playing the same songs in the same styles. Our work demonstrates the possibility of learning causal factors that directly influence the mutual interaction of improvisors. This work and extensions of it might be combined with other computational models of jazz improvisation, including models that make different assumptions about the problem (such as allowing “free” melodic improvisation) or have stronger generative rules for constructing “rhythm section” parts. This could lead to much richer and more realistic models of improvisation in which mutual influences of performers are appreciated by listeners as a key aspect of the performance.

Chapter 8

Facial and Gestural Expression

So far, our study of expressive and interactive music performance has focused on the auditory aspects of artificial performance (i.e., timing and dynamics). However, musical expression and interaction extend beyond sound. Studies [87]-[89] have shown that musicians communicate with each other via not only sound but also visual cues, such as finger movements, facial expressions, and body gestures. In this chapter, we focus on the non-acoustic response of the artificial performer. In particular, we created a music-robotic system (Figure 39) capable of performing an accompaniment for a musician and reacting to human performance with gestural and facial expression in real time.

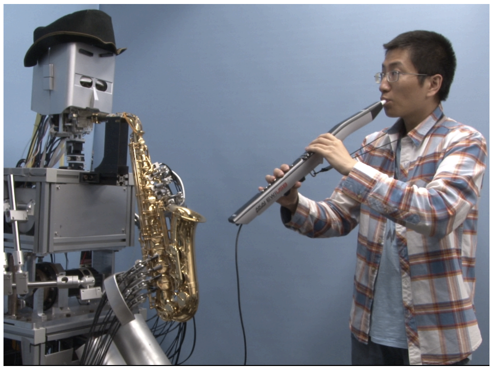


Figure 39. Automatic accompaniment with robot expression.

This music-robotic system bridges and benefits two existing fields: *automatic accompaniment* and *social robotics*. On one hand, automatic accompaniment systems (introduced in Section 5.1) have been developed to serve as virtual musicians capable of performing music with humans. After the invention of the first systems [2][3] in 1984, which used simple models to anticipate the tempo of a monophonic input, many studies extended the model to achieve more expressive music interactions. These extensions include polyphonic [5] and embellished [4] input recognition, smooth tempo adjustment [9][90], and the techniques presented from previous chapters that enable expressive reaction with music nuance [91]. While most efforts focused on the system's auditory aspects, very few models have considered the virtual musician's gestural expression, and no models considered facial expressions. On the other hand, social robots have been developed to interact with humans or other agents following certain rules of social behaviors. Many studies have shown that robot expression, especially humanoid expression, significantly increases the engagement and interaction between humans and computer programs in many forms, such as telecommunication [92] and dialog systems [93]. However, music interaction, as high-level social communication, has not been paid much attention in this context. Though we have seen the development of several music robots, none are able to react to other musicians with human-like expression yet.

We saw that automatic accompaniment and social robotics can complement each other and therefore created the first automatic accompaniment system with humanoid robot expression. To be specific, we integrated the saxophonist robot developed at Waseda University into a software framework for automatic accompaniment. The system currently takes a human musician's MIDI flute performance as input and outputs acoustic accompaniment with gestural and facial expression. The (larger scale) gestural expression reacts to macro-level tempo variations while the (smaller scale) facial expression reacts to micro-level tempo variations. Of course, our first integration does not consider *all* aspects of gestural and facial expression. The current solution considers body and eyebrow movements, and we believe that other aspects of expression can be processed in a similar way. Also, we have not yet incorporated the advanced computational models considering expressive timing and dynamics (described from Chapter 3 to Chapter 5) into our music-robot system.

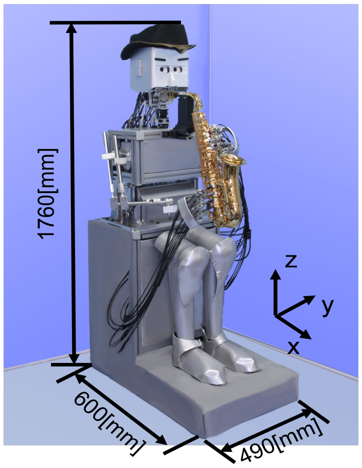
In addition, we conducted subjective evaluations of this integration on audiences to validate the joint effects of robot expression and automatic accompaniment. Our hypothesis is that with humanoid robot expression, an automatic accompaniment system provides more musical, interactive, and engaging performance between humans and machines. We showed video clips in different conditions (with/without expression, with/without accompaniment) to audiences and used repeated-measure ANOVA to measure the difference between different conditions. Our results show that robot embodiment, especially facial expression, improves the subjective ratings on automatic accompaniment significantly. Counterintuitively, such improvement does not exist when the machine is playing a fixed media performance, in which the human musician simply follows the machine that plays a pre-recorded performance.

The rest of this chapter is organized as follows. In Section 8.1, we present the design of the saxophone robot with a focus on its control of body and eyebrow movements. In Section 8.2, we review the automatic accompaniment framework (which was introduced in Section 5.1) and focus on the mapping from MIDI performance to robot motions. In Section 8.3, we present the subjective evaluations and the experimental results.

8.1 The humanoid saxophonist player robot

8.1.1 Waseda saxophonist robot (WAS)

The development of Waseda Saxophonist Robot (WAS) [94][95] was started in 2008. The robot was designed with a critical focus on the physiology and anatomy of the human organs involved during saxophone playing. The fourth version of the robot (WAS-4) was completed in 2015. Face and trunk mobility have been increased, to add basic interaction abilities during artistic joint performances with human partners. During joint musical performances, in fact, musicians cannot use vocal signs and must rely on non-verbal body communication for synchronization. The robot is now able to perform human-like non-verbal signaling, giving partner human players real-time cues on its interpretation, allowing for a better control over synchronization and improving the interaction experience as well as the overall joint musical performance. Figure 40 shows the general design of the robot used in this study.



Function	Body Parts	DoFs
Sound production	Lips	2
	Oral cavity	1
	Tongue	1
	Lung	Pump 1 Valve 1
Key stroke	Fingers	Left 8 Right 11
Body movement	Hip	1
Facial expression	Eyebrow	1
Total		29

Figure 40. An overview of the design of WAS-4.

8.1.2 Body and eyebrow movements

The two interactive movements used in this study are: swinging the upper body and raising/frowning eyebrows. The body positions during a swing movement are shown in Figure 41, where the robot starts from a neutral position (left), swings forward and backward (middle two snapshots), and finally comes back to the neutral position again (right). Eyebrow movements are illustrated in Figure 42, where the left one is neutral, the middle one is raised, and the right one is frowning.

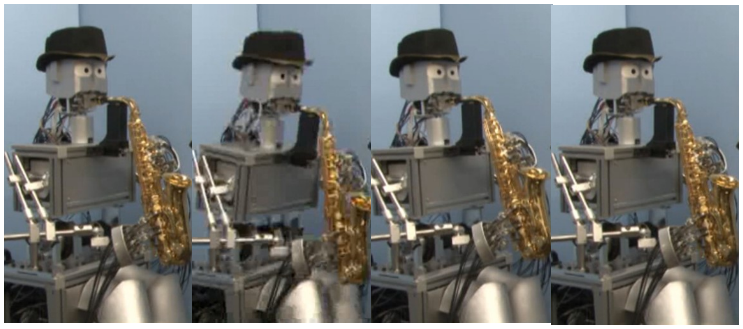


Figure 41. An illustration of the body movement.

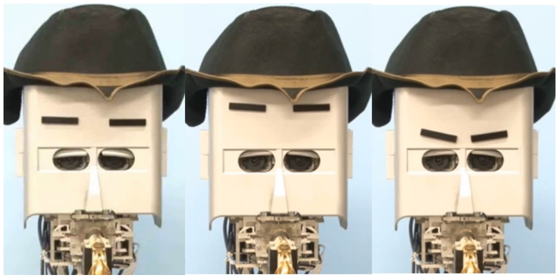


Figure 42. An illustration of the eyebrow positions.

8.2 Automatic accompaniment with robot expression

This section describes how the robot reacts to human performance. There are three main steps: score matching, tempo estimation, and the mapping from tempo to robot expression. The logic flow is shown in Figure 43. Again, the current system takes a human's monophonic MIDI flute performance as input and outputs acoustic accompaniment with eyebrow and body swing movements.

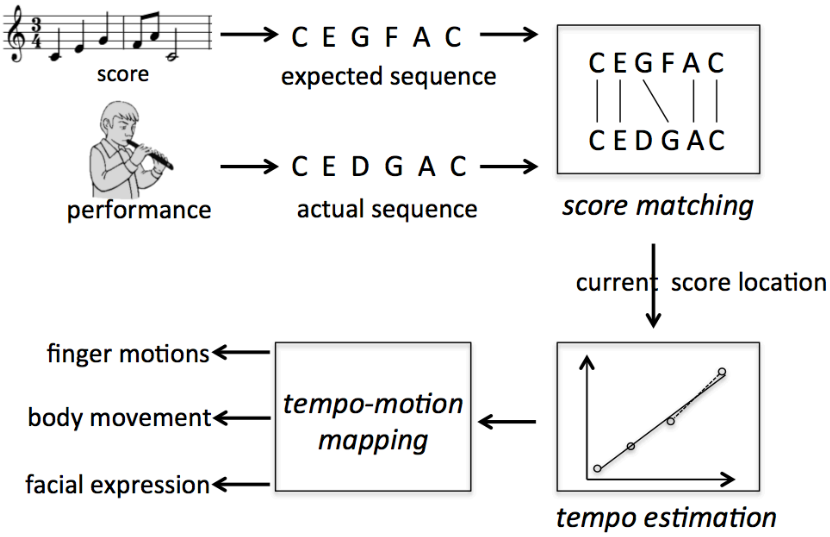


Figure 43. A system diagram of automatic accompaniment system with robot expression.

8.2.1 Score matching and tempo estimation

We used almost the same methods introduced in the baseline computational model (Section 5.1) for performance-score matching and tempo estimation. Let's first review the notations we used. The matched performance notes are denoted by $m = [m_1, m_2, \ldots, m_i, \ldots]$, their performance timings are denoted by $p = [p_1, p_2, \ldots, p_i, \ldots]$, and their score timings are denoted by $s = [s_1, s_2, \ldots, s_i, \ldots]$.

For the matching problem, the only difference here is that the monophonic human performance is generated via a MIDI flute instead of a MIDI piano. For the tempo estimation problem, besides $v = [v_1, v_2, \ldots, v_i, \ldots]$, (i.e., the estimated "4-beat tempo") we estimate a note-wise tempo $v' = [v_1', v_2', \ldots, v_i', \ldots]$, which is more sensitive to instantaneous and local tempo variations. Formally,

$$v_i' = \begin{cases} (s_i - s_{i-1})/(p_i - p_{i-1}), & i > 1 \\ 1, & i = 1 \end{cases} \quad \textbf{(55)}$$

In this chapter, refer to v' as the *micro-scale tempo* and v as the *macro-scale tempo.*

8.2.2 Mapping from performance tempo to robot motions

We designed rule-based methods to control the robot motion by the estimated tempo. The current system separates robot motions into three groups: finger motions which are controlled by macro-scale tempo, body movements which are controlled by the deviation of macro-scale tempo, and eyebrow movements which are controlled by the deviation of micro-scale tempo. These rules are designed according to domain knowledge of music performance.

Finger Motions: Finger motions control the accompaniment, whose timings are specified in another pre-defined score that is synchronized with the score for human performance. The robot uses the latest macro-scale tempo estimation and extrapolates this tempo (slope) to estimate and schedule the next note. It is important to notice that finger motions need high timing accuracy, but robot mechanics has unavoidable latency. To overcome the latency, we schedule the notes ahead of their estimated onset times. Generally, if the latency for the MIDI flute is l_1 and the latency for the robot fingers is l_2, notes whose estimated time are t will be scheduled to execute at $t' = t - (l_2 - l_1)$. In practice, t' is around 50 milliseconds.

Body movements: Body movements are controlled by the deviation of the macro-scale tempo. If the two latest estimated macro-scale tempi both speed up/slow down beyond a certain threshold, a body movement is triggered. By referring to the notations in Section 8.2.1, for $i > 1, j = 0$ and 1, a body movement is triggered if:

$$\left|\frac{v_{i-j} - v_{i-j-1}}{v_{i-j-1}}\right| > p \qquad \textbf{(56)}$$

The rationale of this rule is that performers often use body movements to indicate smooth tempo changes. The current system sets p = 5%. Besides this rule, we also insert a body movement at the beginning and the ending of the robot performance.

Eyebrow motions: Eyebrow motions are controlled by the deviation of the micro-scale tempo. If the two latest estimated micro-scale tempi both slow up beyond a certain threshold, both eyebrows will raise. Similarly, if the tempi speed up beyond a certain threshold, a *frown* motion is triggered. If none of these two conditions are met, eyebrows stay at the neutral position. Formally, for $i > 1$, and $j = 0$ and 1,

$$\text{eyebrow motion} = \begin{cases} \text{raise}, & \text{if } \frac{v'_{i-j} - v'_{i-j-1}}{v'_{i-j-1}} < -q \\ \text{frown}, & \text{if } \frac{v'_{i-j} - v'_{i-j-1}}{v'_{i-j-1}} > q \\ \text{neutral}, & \text{otherwise} \end{cases} \qquad (57)$$

The rationale of this rule is that eyebrow motions are often associated with sudden tempo changes. The current system sets q = 5%. Note that eyebrow motions will be more frequent than body movements under the same threshold because micro-scale tempi are more sensitive.

8.3 Subjective evaluation

We conducted subjective evaluations on audiences to validate the effects of robot expression. We first inspected whether the robot helps with automatic accompaniment. Then, we inspected whether the robot helps with fixed media performance (in which the robot plays a fixed performance and the human performer has to adapt to the robot). Finally, we compared these two results to see the joint effect of robot expression and automatic accompaniment.

8.3.1 The robot effect on automatic accompaniment

Our hypothesis is that with humanoid robot expression, the automatic accompaniment system provides more musical, interactive, and engaging performance between humans and machines. To test this claim, we recorded videos of human-computer interactive performances (of the same piece of music) in 3 different conditions of robot embodiment and invited audiences to provide subjective ratings on these videos.

Video recording setup: The videos were recorded as shown in Figure 44, with the human performer and the robot standing opposite to each other. Figure 39 shows a corresponding screen shot.

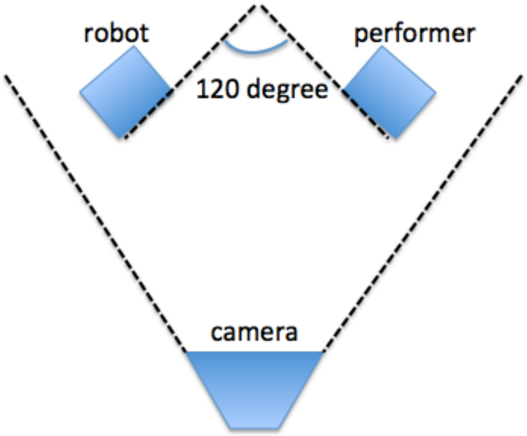


Figure 44. The video recording setup.

Conditions of robot embodiment: The 3 performance conditions are listed in Table 4, where higher index corresponds to greater functionality of the robot. Note that in condition A (blocked robot), we put a cover in front of the robot so that neither the human performer nor the camera could see the robot. The purpose was to block the visual cues but retain the same sound source. In condition B (static body), the robot's body and eyebrows do not move; the only working parts are the mouth and fingers. In condition C (full expression), the robot moves body and eyebrows.

Table 4. The conditions for robot setting.

Index	Robot setting
A	Blocked robot
B	Static body
C	Full expression

The survey: We showed the recorded performance videos in all 3 conditions in a random order to each audience subject without directly revealing the condition. Each video is about 80 seconds long. After each video, audiences were asked to rate the performance based on a 5-point Likert scale from 1 (very low) to 5 (very high) according to three criteria:

- *Musicality:* How musical the performance was.
- *Interactivity:* How close the interaction was between the human performer and the machine.
- *Engagement:* How engaged the human performer was.

Hypothesis test: The null hypothesis is that different conditions have no effect on automatic accompaniment and therefore the ratings under different conditions are the same. Formally:

$$H_0: \quad \mu_A = u_B = \mu_C \tag{58}$$

Similarly, the alternative hypothesis is that:

$$H_1: \quad \exists i, j \in \{A, B, C\}: \mu_i \neq \mu_j \tag{59}$$

Since all the subjects experienced all the conditions, we used within-subject ANOVA [48] (also known as repeated measurement study) to compute the mean standard error (MSE) and p-value. (We use the Huynh-Feldt correction [48] when the sphericity of the data is not met.)

Experimental results: A total of *n* = 33 subjects (14 female and 19 male) have completed the survey. The aggregated result (as in Figure 45) shows that the robot effect improves the subjective ratings of automatic accompaniment.

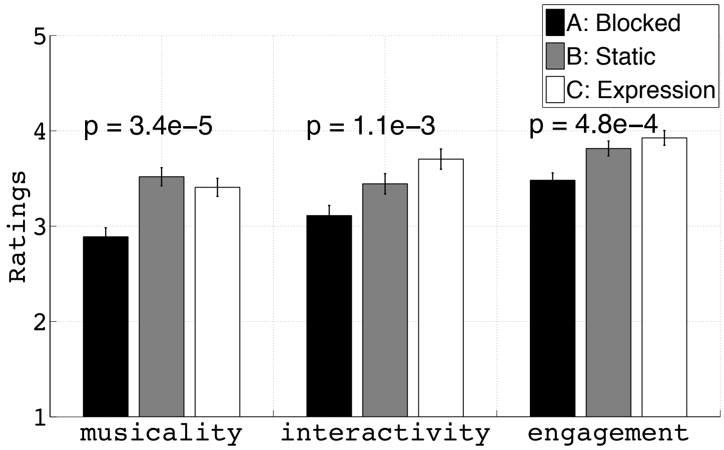


Figure 45. The subjective evaluation results of the robot effect on automatic accompaniment.

Here, different colors represent different conditions. The heights of the bars represent the means of the ratings and the error bars represent the MSEs. It is clear that the robot embodiment and expression improves the ratings and such improvements are monotonic (except for *musicality*) when the functionality of the robot increases. For all three criteria, the p-values are much smaller than 0.005 and hence the improvements are statistically significant.

Main result 13: *For automatic accompaniment, humanoid robot embodiment and expression lead to more musical, interactive, and engaging performance when compared with acoustic accompaniment with no visual contacts with the robot.*

8.3.2 The robot effect on fixed media performance

In addition to the robot effect on automatic accompaniment, we also inspected whether the robot helps with fixed media performance. In this case, the robot played a pre-recorded performance and the human musician adapted to the robot. Similar to Section 8.3.1, the null hypothesis is that different conditions have no effect on the subjective ratings of fixed media performance. With exactly the same video recording setup, conditions of robot setting, and survey process, the result (as in Figure 46) shows that robot embodiment and expression do not help with fixed media performance.

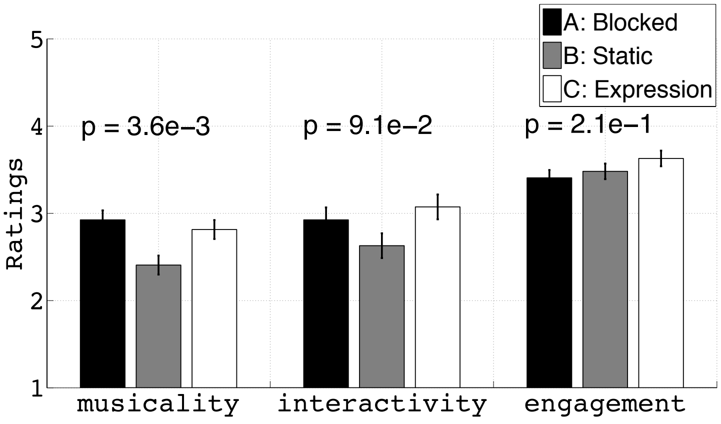


Figure 46. The subjective evaluation results of the robot effect on fixed media performance.

Counterintuitively, for musicality the robot decreases the ratings with the p-value smaller than 0.005. For interactivity and engagement, though we see evidence of improvement, the associated p-values are both larger than 0.05.

Main result 14: *For fixed media performance, humanoid robot embodiment and expression does not lead to more musical, interactive, and engaging performance when compared with acoustic accompaniment with no visual contacts with the robot.*

8.3.3 A comparison between automatic accompaniment and fixed media performance

We finally inspect the joint effect of automatic accompaniment and the robot effect by putting the results of Figure 45 and Figure 46 together, as shown in Figure 47.

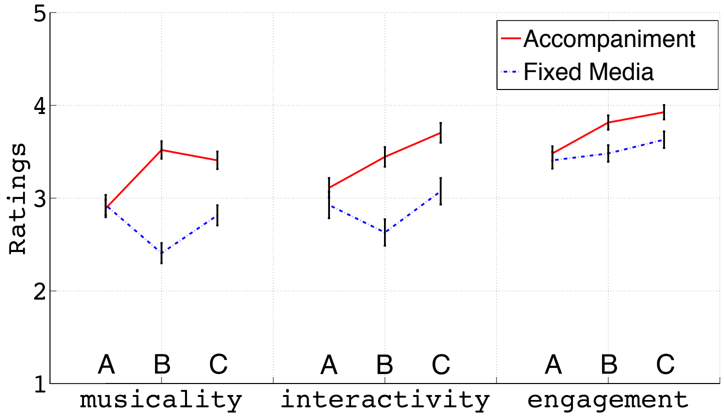


Figure 47. The joint effect of automatic accompaniment and robot expression

For all 3 criteria, the advantage of automatic accompaniment over fixed media in conditions B and C is more significant compared with in condition A. In other words, robot expression amplified the difference between automatic accompaniment and fixed media performance. This result indicates that when we combine the factors of automatic accompaniment and social robotics, music performance between the human and the machine become more musical, interactive, and engaging.

Main result 15: *Humanoid robot embodiment and expression make the difference between automatic accompaniment and fixed media human-computer music performance more pronounced.*

8.4 Discussion

In conclusion, we have combined the efforts of social robotics and automatic accompaniment to create an automatic accompaniment system with humanoid robot expression. As far as we know, this is the first interactive music performance between a human musician and a humanoid music robot with systematic subjective evaluation. Our result shows that expressive humanoid robots lead to more musical, interactive, and engaging automatic accompaniment when compared with acoustic accompaniment with no robot effect. Counterintuitively, this improvement does not exist for fixed media performance.

This study contributes to the computer music community by providing the first subjective evaluation on the joint effect of automatic accompaniment and robot expression. It also contributes to the social robotics community by proving that the benefits of humanoid robots generalize to the interactive music performance scenario. The result shows the benefit of combining interactive computer music systems with humanoid robots, which points to the integration of these two fields for future research.

Chapter 9

Conclusion

9.1 Summary

Techniques of Artificial Intelligence and Human-Computer Interaction have empowered computer music systems with the ability to perform with humans via a wide spectrum of applications, ranging from fixed media performance to free improvisation. However, the musical interaction between humans and machines is still far less musical than the interaction between human musicians, since most systems lack representations and capabilities of *musical expression*. This thesis contributes various computational techniques, especially machine-learning algorithms, to create artificial musicians that perform *expressively and collaboratively* with humans based on the current framework of automatic accompaniment systems (which also serve as our baseline). In particular, the thesis focuses on 1) expressive timing and dynamics, 2) basic improvisation techniques, and 3) facial and body gestures in collaborative performance.

Timing and dynamics are the two most fundamental aspects of musical expression and also the main focus of this thesis. They are studied in an expressive piano duet setting. We contribute the first dataset of expressive piano duets, which contains 10 pieces of music, each with 12 to 35 performances. In addition to the performance data, we contribute a general feature scheme to represent collaborative performance from various aspects of music context and also contribute a method that derives chord features from the features of individual notes within the chord based on perceptual evidence. As for the learning task, we model the expression of two pianists as co-evolving time series and develop the first set of algorithms that discover the regularities of expressive musical interaction from rehearsals. Based on the learned models, an artificial pianist generates its own musical expression by interacting with a human pianist given a pre-defined score.

The results show that, given a small number of rehearsals, we can successfully apply machine learning to generate more expressive and human-like collaborative performance than the baseline. This is the first application of spectral learning in the field of music. The result is supported by different measurements:

- In terms of the average absolute difference between machine prediction and human performance, our method shrinks the error as much as 50 milliseconds for timing and 8 MIDI velocity units for dynamics.
- In terms of the average absolute asynchrony and dynamics difference between the two piano performances, the errors (between human performance and our method) are only about 10 milliseconds for timing and 2 MIDI velocity units for dynamics.
- In terms of the Kolmogorov–Smirnov distance between machine generation the human performance, our model shrinks the distance as much as 0.35 for timing and 0.2 for dynamics compared with the baseline.
- In terms of subjective ratings where higher ratings mean better results, our method is higher than the baseline by 0.9 units based still lower than the human performances by 0.5 units based on a Likert scale from 1 to 5.

Along with the results, we have several discoveries related to musical expression in piano duets. Some important discoveries are:

- The prediction of expressive timing is almost exclusively related to the timing of other nearby notes and is especially dependent upon the rhythmic context, while the prediction of expressive dynamics is related to all aspects of music context.
- For both expressive timing and dynamics, better prediction than the baseline can be achieved as long as the learning algorithms look ahead for a phrase or even 3 to 4 notes in rehearsals.
- There exists a latent expressive space, which explains a significant portion of musical expression. The dimensionality of this latent space is only 4 to 7.

Last but not least, we have explored different training strategies under realistic constraints of data collection. The experimental results of different strategies indicate that there are multiple factors that determine expressive performance:

- Training on consistent performances or rehearsals by the *same* pair of performers leads to better results than randomly sampling the rehearsals. This result indicates expressive musical expressive is partially determined by performance styles and performer preferences.

- Even trained on rehearsals of *different* pieces of music, our method can still outperform the baseline. This result indicates expressive musical interaction is partially determined by general musicianship and universal rules.

Besides expressive timing and dynamics, we consider some basic improvisation techniques in a piano duet setting where musicians have the freedom to interpret pitches and rhythms. The study of the improvisation aspect of musical expression is the most recent effort of the thesis. We collected another duet dataset, which contains two songs, each performed 15 times by the same pair of musicians. The first piano part is still a monophonic melody, but the performer now has the freedom to add embellishments (such as trills, mordent, etc.) given a pre-defined score. The second piano part no long follows a score but a lead sheet, which only specifies a sequence of chords, leaving the realization of pitches and rhythms to the performer. We developed a measure-specific approach, which trains a different model for each individual measure, and focused on the prediction of the numbers of onsets and notes for each measure. Given the model prediction, we generate the improvised score using *nearest-neighbor search*, which selects the training example whose onset and notes numbers are closest to the estimation. Our result shows that the developed measure-specific approach generates better result than the baseline. The experimental evidence includes:

- On average, our model can shrink the difference between machine prediction and human performance as much as 0.2 units (for the number of onsets) and 0.4 units (for the average number of notes per chord) compared with the baseline.
- In terms of subjective ratings where higher ratings mean better results, our model improves the rating by about 1 unit (in terms of a Likert scale from 1 to 5) compared with the baseline.

Musical expression generally means only aspects of musical performance that produce sound. In many concert situations, physical gestures including facial expression and body movements are important to the perception of music. These aspects of musical expression were studied using a humanoid saxophonist robot from Waseda University. Based on the current framework of automatic accompaniment, we contributed the first algorithm to enable a robot to perform an accompaniment for a musician and react to human performance with gestural and facial expression. The current system uses rule-based performance-motion mapping and separates robot motions into three groups: finger motions which are controlled by macro-scale tempo, body movements which are controlled by the deviation of macro-scale tempo, and eyebrow movements which are controlled by the deviation of micro-scale tempo. We also conducted the first subjective evaluation on the joint effect of automatic accompaniment and robot expression.

Our result shows that when we combine the factors of automatic accompaniment and social robotics, music performance between the human and the machine is perceived to be more musical, interactive, and engaging. Experimental evidence which support this conclusion includes:

- For automatic accompaniment, humanoid robot embodiment and expression leads to higher subjective ratings on musicality, interactivity, and human performer engagement when compared with acoustic accompaniment with no visual contacts with the robot.
- For fixed media performance, humanoid robot embodiment and expression only *decrease* the subjective ratings when compared with acoustic accompaniment with no visual contacts with the robot.

9.2 Limitations

For note-level musical expression, the expressive timing discussed in this thesis has not yet included note *duration*, which is also an important element of expression especially for "*staccato*" performances in a musical term. The synthesized artificial performances used in this thesis simply adopted the note durations by averaging the rehearsals (for same-piece learning task) or from scores (for cross-piece learning task). We actually tried to learn durations using the same learning techniques but saw that note durations are too noisy to learn possibly due to the sustain pedal effect.

For larger-scale musical expression, this thesis has not yet considered any music structure beyond the scope of a *phrase*. So far, this limitation has not caused any problem because the music pieces we used are all very short (within 2 minutes). However, if we want to learn longer pieces with complex music structure, such as a sonata or a concerto, features of larger-scale music structure will presumably make more differences.

For musical collaboration, the current system is not yet a "closed-loop system" because we has not modeled how human musicians change their behavior in response to machine behavior. All the synthesized artificial performances for evaluation purpose were still generated in "single-blind" simulations, where only the machines can "hear" humans. To close the loop, we should turn the current "batch" learning algorithms into online learning, especially reinforcement learning algorithms.

9.3 Future work

Future work aims to empower intelligent systems with more profound artificial musicianship to master musical expressive in collaborative performance through wider interdisciplinary efforts. Such systems will be able to serve people not only through music performance but also through music education and music therapy. This section outlines three future directions; each requiring a progressively deeper understanding of music and providing a larger potential impact on people's daily lives.

More advanced interactive music robots: The current music robot only uses facial and gestural expression as an output; it is still blind and does not react to the visual cues of human performers. In addition, the system is still rule-based. In the future, we are going to place cameras on the robot, use motion-capture systems to collect rehearsal videos, and apply machine-learning algorithms to integrate different musical expressions for robots. A robot musician can serve as a personal music partner. We can further generalize this idea and imagine a personal robot orchestra, with which even amateur musicians can hold solo concerts easily. *This is one possible future direction for music performance*.

More advanced learning methods for taste in music: One step further toward a profound musical intelligence is *taste*. An intelligent system with a taste in music will be able to train itself by learning from expressive examples selectively, rather than through a passive and fully supervised procedure. By using semi-supervised learning, especially active learning, a system can learn basic musical expression from a small number of labeled human performances and then improve itself automatically through its own experience or a vast amount of unlabeled data collected by music information retrieval techniques. Self-trained systems will require minimal programming and calibration from humans. They will be able to use feedback to adapt their expressive performances to different performer preferences, performance styles, instrument qualities, and acoustic environments. A special application is to just "copy" a performance from a concert hall and later adaptively "paste" it to our home. In other words, we can listen to "live" symphonies at home using a robot orchestra. *This is one possible future direction for music appreciation*.

Music education and therapy: Today, humans design algorithms to develop artificial performers; tomorrow, machines can help teach music to humans. An artificial music teacher can give us feedback at any time, making music training a lot easier and potentially much cheaper. In addition, human-robot musical interaction enables us to explore how different teaching strategies affect the way students learn, since the robot behaviors can be easily configured by a set of parameters. Each step in music training has three components: 1) an accurate judgment of the current level of expressive performance, 2) a reachable next-level target, and 3) a tailored plan to reach that target. Though solving all three problems autonomously is a long-term goal, we can combine machine learning with *human computation* (crowdsourcing) to design a music education curriculum jointly with machines. My vision for unifying music education and therapy is inspired by *Eurhythmics,* a traditional music training method that focuses on the intrinsic relationship between body movements and musical expression. For example, a Eurhythmics instructor plays a tricky music segment on the piano; students are asked to step on the downbeats while clapping on the upbeats to show their mastery of a certain rhythm. This procedure (of training rhythmic feeling) can be turned easily into an expressive and collaborative performance, where intelligent systems can play the piano part while evaluating the movements of the students. Moreover, this method can be adapted to physical therapy. Compared with current approaches in physical therapy for Parkinson's disease where doctors still use metronomes to help patients recover their ability to walk smoothly, an interactive process involving musical expression will be a huge improvement.

References

[1]. Hugo, V. (2001). *William Shakespeare*. The Minerva Group, Inc.

[2]. Dannenberg, R. (1985). An On-Line Algorithm for Real-Time Accompaniment. *Proceedings of the 1984 International Computer Music Conference*, 193-198.

[3]. Vercoe, B. (1985). The Synthetic Performer in the Context of Live Performance. *Proceedings of the 1984 International Computer Music Conference*, 199-200.

[4]. Dannenberg, R. and Mukaino, H. (1988). New Techniques for Enhanced Quality of Computer Accompaniment. *Proceedings of the International Computer Music Conference*, 243–249.

[5]. Bloch, J. and Dannenberg, R. (1985). Real-Time Accompaniment of Polyphonic Keyboard Performance. *Proceedings of the International Computer Music Conference*, 279-290.

[6]. Grubb, L. and Dannenberg, R. (1994), Automated Accompaniment of Musical Ensembles. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 94-99.

[7]. Grubb, L. and Dannenberg, R. (1997). A Stochastic Method of Tracking a Vocal Performer. *Proceedings of the International Computer Music Conference*, 301-308.

[8]. Raphael, C. (2010). Music Plus One and Machine Learning Machine Learning, *Proceedings of the Twenty-Seventh International Conference on Machine Learning, ICML.*

[9]. Cont, A. (2008). ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters. Computer Music. *Proceedings of International Computer Music Conference*.

[10]. Large, E. W. and Palmer, C. (2002). Perceiving Temporal Regularity in Music. *Cognitive Science*, 26, 1–37.

[11]. Cont, A. (2006). Realtime Audio to Score Alignment for Polyphonic Music Instruments Using Sparse Non-negative Constraints and Hierarchical HMMs. *Proceedings of the IEEE International Conference on Acoustics and Speech Signal Processing (ICASSP)*.

[12]. Desain, P., and Honing, H. (1994). Does Expressive Timing in Music Performance Scale Proportionally with Tempo? *Psychological Research*, 56(4), 285-292.

[13]. Kirke, A., and Miranda. E. R. (2009). A Survey of Computer Systems for Expressive Music Performance. *ACM Surveys* 42(1): Article 3.

[14]. Widmer, G. and Goebl, W. (2004). Computational Models of Expressive Music Performance: The State of the Art. *Journal of New Music Research*., 203 -216.

[15]. Sundberg, J., Askenfelt, A., and Fryden, L. (1983). Musical Performance: A Synthesis-by-rule Approach. *Computer Music Journal*. 7, 37–43.

[16]. Friberg, A. (1991). Generative Rules for Music Performance. *Computer Music Journal*, 15, 56–71.

[17]. Friberg, A., Frydén, L., Bodin, L., and Sundberg, J. (1991). Performance Rules for Computer-Controlled Contemporary Keyboard Music. *Computer Music Journal*, 15, 49–55.

[18]. Sundberg, J., Friberg, A., and Bresin, R. (2003). Attempts to Reproduce a Pianist's Expressive Timing with Director Musices Performance Rules. *Journal of New Music Research*, 32, 317–325.

[19]. Todd, N.P.M. (1985). A Model of Expressive Timing in Tonal Music. *Music Perception*, 3, 33–58.

[20]. Todd, N.P.M. (1989). Towards a Cognitive Theory of Expression: The Performance and Perception of Rubato. *Contemporary Music Review*, 4, 405–416.

[21]. Widmer, G. (2003). Discovering Simple Rules in Complex Data: A Meat-learning Algorithm and Some Surprising Musical Discoveries. *Artificial Intelligence*, 146, 129–148.

[22]. Widmer, G. (2002). Machine Discoveries: A Few Simple, Robust Local Expression Principles. *Journal of New Music Research*, 31, 37–50.

[23]. Arcos, J.L., Mántaras, R., López de, and Serra, X. (1998), SaxEx: A Case-based Reasoning System for Generating Expressive Performances. *Journal of New Music Research*, 27, 194–210.

[24]. Widmer, G. and Tobudic, A. (2003). Playing Mozart by Analogy: Learning Multi-level Timing and Dynamics Strategies. *Journal of New Music Research*, 32, 259–268.

[25]. Tobudic, A. and Widmer, G. (2003). Relational IBL in Music with a New Structural Similarity Measure. *Proceedings of the 13th International Conference on Inductive Logic Programming* (ILP'03), Szeged, Hungary, 365–382.

[26]. Flossmann, S., Grachten, M., and Widmer G. (2013). Expressive Performance Rendering with Probabilistic Models. *Guide to Computing for Expressive Music Performance*, A. Kirke and E. Miranda, Eds. Springer, 75–98.

[27]. Kim, T., Satoru F., Takuya, N., and Shigeki, S. (2011). Polyhymnia: An Automatic Piano Performance System with Statistical Modeling of Polyphonic Expression and Musical Symbol Interpretation. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 96-99,

[28]. Grindlay, G., and Helmbold, D. (2006). Modeling, Analyzing, and Synthesizing Expressive Piano Performance with Graphical Models. *Machine learning*, 65(2-3), 361-387.

[29]. Repp, B. (1997). Some Observations on Pianists' Timing of Arpeggiated Chord. *Psychology of Music*, 25(2): 133-148.

[30]. Hoffman, G. and Weinberg, G. (2011). Interactive Improvisation with A Robotic Marimba Player. *Autonomous Robots* 31, No. 2-3, 133-153.

[31]. Sun, S., Trishul, M., and Weinberg G. (2012). Effect of Visual Cues Synchronization of Rhythmic Patterns. *Proceedings of International Conference of Music Perception and Cognition*.

[32]. Xia, G., Tay, J., Dannenberg, R., and Veloso, M. (2012). Autonomous Robot Dancing Driven by Beats and Emotions of Music. *Proceedings of The 11th International Conference On Autonomous Agents and Multiagent Systems*. 1, 205-212.

[33]. Lim, A. et al. (2010). Robot Musical Accompaniment: Integrating Audio and Visual Cues for Real-Time Synchronization with A Human Flutist. *Proceedings of the IEEE/RSJ International Conference*, 1964-1969.

[34]. Otsuka, T. et al. (2009). Incremental Polyphonic Audio with Score Alignment Using Beat Tracking for Singer Robots. *Proceedings of the IEEE/RSJ International Conference*, 2289-2296.

[35]. Weinberg, G., Driscoll, S., and Parry, M. (2005). Musical Interactions with A Perceptual Robotic Percussionist. *IEEE International Workshop*, 456-461.

[36]. Kapur, A. (2011). Multimodal Techniques for Human/Robot Interaction. *Musical Robots and Interactive Multimodal Systems*, Springer Berlin Heidelberg, 215-232.

[37]. Yule, G. Udny (1927). On A Method of Investigating Periodicities Disturbed Series, with Special Reference with Wolfer's Sunspot Numbers. *Philosophical Transactions of the Royal Society of London*, Ser. A, Vol. 226, 267–298.

[38]. Walker, G. (1931). On Periodicity Series of Related Terms. *Proceedings of the Royal Society of London*, Ser. A, Vol. 131, 518–532.

[39]. Yuan, M. and Lin, Y. (2006). Model Selection and Estimation Regression with Grouped Variables. *Journal of The Royal Statistical Society: Series B (Statistical Methodology*), 68(1), 49-67.

[40]. Friedman, J., Hastie, T., and Tibshirani, R. (2010). A Note on The Group Lasso and A Sparse Group Lasso. Arxiv Preprint Arxiv: 1001.0736.

[41]. Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1), 55-67.

[42]. Van Overschee, P. and De Moor, B. (2012). *Subspace Identification for Linear Systems: Theory, Implementation, Methods*. Springer Science & Business Media.

[43]. Boots, B. (2012). *Spectral Approaches with Learning Predictive Representations* (No. CMU-ML-12-108). Doctoral Thesis. Carnegie-Mellon Univ., School of Computer Science.

[44]. Boots, B., Siddiqi, S. M., and Gordon, G. J. (2011). Closing the Learning-Planning Loop with Predictive State Representations. *The International Journal of Robotics Research*, 30(7), 954-966.

[45]. Boots, B. and Gordon, J. (2011). An Online Spectral Learning Algorithm for Partially Observable Nonlinear Dynamical Systems. *Proceedings of the Twelfth National Conference on Artificial Intelligence*.

[46]. Hefny, A., Downey, C., and Gordon, G. J. (2015). Supervised Learning for Dynamical System Learning. *Advances Neural Information Processing Systems*, 1963-1971.

[47]. Tabachnick, B. G., Fidell, L. S., and Osterlind, S. J. (2001). *Using Multivariate Statistics*.

[48]. Ellen, R. and Girden, E. (1992). *ANOVA: Repeated Measures*. No. 84. Sage.

[49]. Loftus, G. and Masson, M. (1994). Using Confidence Intervals: within-Subject Designs. *Psychonomic Bulletin and Review*, L(4), 476-490.

[50]. Large, E. W. and Kolen, J. F. (1994). Resonance and the Perception of Musical Meter. *Connection Science*, 6, 177–208.

[51]. Repp, B. H. and Keller, P. E. (2004). Adaptation with Tempo Changes Sensorimotor Synchronization: Effects of Intention, Attention, and Awareness. *Journal of Experimental Psychology*, 57A, 499-521.

[52]. Vorberg, D. and Schulze, H. (2002). A Two-Level Timing Model for Synchronization. *Journal of Mathematical Psychology*, 46, 56–87.

[53]. Mates, J. (1994). A Model of Synchronization of Motor Acts with A Stimulus Sequence. *Biological Cybernetics*, 70, 463– 473.

[54]. Large, E. W. and Palmer, C. (2011). Temporal Coordination and Adaptation with Rate Change Music Performance. *Journal of Experimental Psychology: Human Perception and Performance*, 37, 1292-1309.

[55]. Repp, B.H. and Keller, P.E. (2008). Sensorimotor Synchronization with Adaptively Timed Sequences. *Hum. Mov. Sci*. 27, 423-456

[56]. Hove, M. J., Spivey, M. J., and Krumhansl, L. (2010). Compatibility of Motion Facilitates Visuomotor Synchronization. *Journal of Experimental Psychology: Human Perception and Performance*, 36(6), 1525-1534.

[57]. Wing, A. M. (2002). Voluntary Timing and Brain Function: An Information Processing Approach. *Brain and Cognition*, 48, 7-30.

[58]. Schöner, G. (2002). Timing, Clocks, and Dynamical Systems. *Brain and Cognition*, 48, 31-51.

[59]. Bartlette, C., Headlam, D., Bocko, M., and Velikic, G. (2006). Effect of Network Latency on Interactive Musical Performance. *Music Perception*, 24, 49–62.

[60]. Keller, P. E., Knoblich, G., and Repp, B. H. (2007). Pianists Duet Better when They Play with Themselves: on the Possible Role of Action Simulation Synchronization. *Consciousness and Cognition*, 16, 102–111.

[61]. Keller, P. E. (2008). Joint Action in Music Performance. *Emerging Communication*, 10, 205.

[62]. Goebl, W. and Palmer, C. (2009). Synchronization of Timing and Motion Among Performing Musicians. Music Perception, 26, 427– 438.

[63]. Galway, J. and Coulter, P. (1997), *Legends*, Hal Leonard.

[64]. Galway, J. (1980). *Songs for Annie: A Collection of Favorite Encores: for Flute with Piano Accompaniment*.

[65]. Galway, J. (1977). *Showpieces, Arranged for Flute and Piano*. Novello.

[66]. Goebl, W. (2003). The Role ff Timing Intensity in the Production and Perception of Melody in Expressive Piano Performance. Doctoral Thesis, Institut für Musikwissenschaft, Karl-Franzens-Universität Graz, Graz, Austria.

[67]. Repp, B. (1996). The Art of Inaccuracy: Why Pianists' Errors Are Difficult to Hear. *Music Perception: An Interdisciplinary Journal* 14.2, 161-183.

[68]. Crestmuse. Http://Www.Crestmuse.Jp/Pedb

[69]. Raphael, C. (2004). A Hybrid Graphical Model for Aligning Polyphonic Audio with Musical Scores. *Proceedings of International Society for Music Information Retrieval Conference*.

[70]. Hoshishiba, T., Horiguchi, S., and Fujinaga, I. (1996). Study of Expression and Individuality in Music Performance Using Normative Data Derived From MIDI Recordings of Piano Music. *Proceedings of International Conference on Music Perception and Cognition*, 465-470.

[71]. Repp, B. (1994). Relational Invariance of Expressive Microstructure Across Global Tempo Changes in Music Performance: An Exploratory Study. *Psychological Research*, 56.4: 269-284.

[72]. Fu, M., Xia, G., Dannenberg, R., and Wasserman, L. (2015). A Statistical View on the Expressive Timing of Piano Rolled Chords. *Proceedings of the International Society for Music Information Retrieval Conference*.

[73]. Fu, M. (2016). *An Analysis of Chord Timing and Dynamics in Expressive Piano Performance*. Masters Thesis, Carnegie-Mellon Univ, School of Music.

[74]. Dannenberg, R. and Wasserman, L. (2009). Estimating The Error Distribution of A Single Tap Sequence without Ground Truth. *Proceedings of the International Society for Music Information Retrieval Conference*.

[75]. Verceo, B. (1985). Synthetic Rehearsal: Training the Synthetic Performer. *Proceedings of the International Computer Music Conference* Proceedings.

[76]. Todd, P. (1989). A Connectionist Approach to Algorithmic Composition, *Computer Music Journal*, 27-43.

[77]. Friberg, A. and Sundberg, J. (1993). Perception of Just- Noticeable Time Displacement of A Tone Presented In A Metrical Sequence At Different Tempos. *The Journal of The Acoustical Society of America*. 94(3), 1859-1859.

[78]. Keller, P., Günther, K., and Repp, B. (2007). Pianists Duet Better when They Play with Themselves: on the Possible Role of Action Simulation In Synchronization. *Consciousness and Cognition* 16.1, 102-111.

[79]. Stephens, M. (1974). EDF Statistics for Goodness of Fit and Some Comparisons. *Journal of the American Statistical Association* 69.347, 730-737.

[80]. Xia, G., Wang, Y., Dannenberg, R., and Gordon, G. (2015). Spectral Learning for Expressive Interactive Ensemble Music Performance. *Proceedings of the 16th International Society for Music Information Retrieval Conference*.

[81]. Rowe, R. (1993). *Interactive Music Systems: Machine Listening & Composing*. MIT Press.

[82]. Chadabe, J. (1984). Interactive Composing: An Overview. *Computer Music Journal* 8.1, 22-27.

[83]. Thom, B. and Dannenberg, R. (1995). Predicting Chords in Jazz. *Proceedings of the International Computer Music Conference*, Banff, Alberta, Canada, 237-238.

[84]. Hiller, L. and Leonard, L. (1979). *Experimental Music: Composition with An Electronic Computer*. Greenwood Publishing Group Inc.

[85]. Thom, B. (2000). Unsupervised Learning and Interactive Jazz/Blues Improvisation. *Proceedings of the Twelfth National Conference on Artificial Intelligence*.

[86]. Kaliakatsos-Papakostas, M., Andreas F., and Michael N. V. (2012) Intelligent Real-Time Music Accompaniment for Constraint-Free Improvisation. *Proceedings of the 24th International Conference on Tools with Artificial Intelligence*.

[87]. K. Katahira et al. (2007). The Role of Body Movement In Co-Performers' Temporal Coordination. *Proceedings of the Inaugural International Conference on Music Communication Science*, 72.

[88]. Fredrickson, W. E. (1994). Band Musicians' Performance and Eye Contact as Influenced by Loss of A Visual and/or Aural Stimulus. *Journal of Research in Music Education*, 42(4), 306-317.

[89]. Kawase, S. (2014). Importance of Communication Cues in Music Performance According to Performers and Audience. *International Journal of Psychological Studies*, 6(2), 49.

[90]. Liang, D., Xia, G., and Dannenberg, R. (2011). A Framework for Coordination and Synchronization of Media. *Proceedings of The International Conference on New Interfaces for Musical Expression*.

[91]. Xia, G. and Dannenberg, R. (2015). Duet Interaction: Learning Musicianship for Automatic Accompaniment. *Proceedings of The International Conference on New Interfaces for Musical Expression*.

[92]. Adalgeirsson, S. (2010) Mebot: A Robotic Platform for Socially Embodied presence. *Proceedings of The Fifth ACM/IEEE International Conference on Human-Robot Interaction*, 15-22.

[93]. Lee, J.K. and Breazeal, C. (2010). Human Social Response Toward Humanoid Robot's Head and Facial Features. *CHI Extended Abstracts*, 4237-4242.

[94]. Solis, J., Ninomiya, T., Petersen, K., Takeuchi, M., and Takanishi, A. (2010). Development of The Anthropomorphic Saxophonist Robot WAS-1: Mechanical Design of The Simulated Organs and Implementation of Air Pressure Feedback Control. *Advanced Robotics*, *24*(5-6), 629-650.

[95]. Matsuki, K., Yoshida, K., Sessa, S., Cosentino, S., Kamiyama, K., and Takanishi, A. (2016). Facial Expression Design for The Saxophone Player Robot WAS-4. *ROMANSY 21-Robot Design, Dynamics and Control*.