Basis-Function Modeling of Loudness Variations in Ensemble Performance

Thassilo Gadermaier, Maarten Grachten, and Carlos Eduardo Cancino Chacón

Austrian Research Institute for Artificial Intelligence
firstname.lastname@ofai.at

Abstract This paper describes a computational model of loudness variations in expressive ensemble performance. The model predicts and explains the continuous variation of loudness as a function of information extracted automatically from the written score. Although such models have been proposed for expressive performance in solo instruments, this is (to the best of our knowledge) the first attempt to define a model for expressive performance in ensembles. To that end, we extend an existing model that was designed to model expressive piano performances, and describe the additional steps necessary for the model to deal with scores of arbitrary instrumentation, including orchestral scores. We test both linear and non-linear variants of the extended model on a data set of audio recordings of symphonic music, in a leave-one-out setting. The experiments reveal that the most successful model variant is a recurrent, non-linear model. Even if the accuracy of the predicted loudness varies from one recording to another, in several cases the model explains well over 50% of the variance in loudness.

Keywords: Musical Expression, Computational Modeling, Neural Networks, Ensemble Performance

1 Introduction

This paper describes a computational model of loudness variations in ensemble performance. We are primarily interested in the expressive factors that influence loudness. Although expression is a very broad term that may include the mental or physical state of the performer(s), their communicative intentions, the targeted audience, and so on, we focus on those aspects of expression that are determined by the written score. In other words, the proposed model is intended to account for the ways in which information extracted from the written score influences the continuous variation of loudness throughout the recording of a performance.

The potential uses of such a model are twofold. Firstly, its predictive capacities can be used to generate more natural, musically appropriate acoustic renderings of a piece, than a straight-forward mechanical rendering. Such improved renderings can also improve tasks such as offline score-performance alignment,

and *automatic live score-following*—a scenario in which a computer keeps track of musical time during the performance of a piece of music [1].

Secondly, a model of expressive loudness variations may also be used for explanatory purposes. This means that the model can attribute variations in the expressive quality of a performance to factors like performance directives that were written in the score by the composer (like *crescendo*, *diminuendo*, and *fermata*), and other aspects of the written score. Explanatory visualizations of expressive performances based on this information can be used for didactic purposes, to introduce an audience to the phenomenon of expressive music interpretation.

As the point of departure for the model of expressive ensemble performance proposed here, we take an existing framework, the *basis-function* modeling approach, which has been successfully used in modeling solo piano [2]. In addition to this linear version of the model, improved results have been obtained using non-linear variants. The non-linear variants can model more complex relationships between expressive parameters and the score, as demonstrated in [3], where the basis-function representation is combined with a *feed-forward neural network* (FFNN). A more sophisticated form of non-linear modeling involves *recurrent* network connections, allowing for temporal dependencies in the relation between score information and expressive parameters. This type of model was shown to outperform non-temporal models for predicting expressive timing in classical piano performances [4]. In the current paper, we employ both the linear, and the two non-linear variants of the model.

The main contribution of the current paper is the extension of the basis-function definition to accommodate for ensembles of instruments, possibly including multiple instances of the same instrument, as is common in orchestral scores. We discuss the difficulties and complications that arise when dealing with recordings of large ensembles, rather than a single piano. To address these issues, we define *merging* and *fusion* operations on basis-function representations, as explained further on in the paper. These operations are needed to train the model on pieces with different instrumentations, and to present the score information of the joint orchestral score to the model in a unified way.

We evaluate the proposed basis-function model for ensemble performance using a dataset of 16 orchestral recordings of pieces by Bruckner, Mahler, and Beethoven, as performed by the Royal Concertgebouw Orchestra. The results show that depending on the piece, a considerable part of the total variance in loudness can be explained by information from the score. Furthermore, there are notable differences between the models, with the non-linear models, especially the recurrent model, performing better than the simpler linear model.

In Section 2, we give a very brief overview music expression research, focusing on computational approaches. Section 3 covers the description of the proposed model for ensemble performance. An experimental validation of the model is described in Section 4, including the presentation and discussion of results. Finally, conclusions are formulated in Section 5.

2 Related work and state of the art

Empirical research and modeling of musical expression have a long history, with accounts of measurements of music performances dating back as far as the late 19^{th} century [5], and the first half of the 20^{th} century [6]. Despite these early precursors, expressive performance research has gained substantial traction only since the 1980s, presumably incited in part by the advent of modern computers, electronic instruments, and the corresponding MIDI protocol for transmission of musical information, facilitating the recording of performances, and subsequent analysis of the data obtained in this way.

A significant number of empirical studies have sought to establish relationships between some aspects of expression and particular explanatory factors. These factors can be roughly divided into those that relate to the performer's intention of expressing particular emotions, and those that relate to the musical structure, in the broadest sense of the word. For example, a widely confirmed mapping between emotion and expression is that slow tempo, legato articulation, and softer timbres contribute to the perception of the music as sad or solemn, whereas a faster tempo, staccato articulation and brighter timbres tend to induce a perception of happiness [7], [8], [9]. Similarly, various structural aspects of the musical score have been found to influence musical expression [10]. Most notably, musical grouping structure (the division of the music into *motifs*, and *phrases*) is often expressed in arc like shapes in tempo and dynamics [11]. Another type of musical structure that musicians express through expressive variations is the metrical structure [12].

Research on expression in ensemble performance is sparse. Studies in this area often focus on synchronization between musicians [13], [14], and the cues musicians use to communicate and synchronize [15], [16].

2.1 Computational modeling of musical expression

Computational models of expressive music performance seek to clarify the relationships between certain properties of the musical score and performance context with the actual performance of the score [17]. These models can serve mainly analytical purposes [18,19], by showing the relation between structural properties of the music and its effect in the performance of such music, mainly predictive purposes [20], i.e. the models are used to render expressive performances, or both [21], [22,2]. Computational models of music performance tend to follow two basic paradigms: *rule based* approaches, where the models are defined through music-theoretically informed rules that intend to map structural aspects of a music score to quantitative parameters that describe the performance of a musical piece, and *data-driven* (or *machine learning*) approaches, where the models try to infer the rules of performance from analyzing patterns obtained from (large) datasets of observed (expert) performances [23].

One of the most well-known rule-based systems for musical music performance was developed at the Royal Institute of Technology in Stockholm (referred to as the KTH model) [24]. This system is top-down approach that describes

expressive performances using a set of (music theoretically sound/cognitively plausible) performance rules that predict aspects of timing, dynamics and articulation, based on a local musical context.

Among the machine learning methods for musical expression is the model proposed by [25]. This model uses artificial neural networks (NNs) in a supervised fashion in two different contexts: 1) to learn and predict the rules proposed by the KTH model and 2) to learn the performing style of a professional pianist using an encoding of the KTH rules as inputs. Similarly, the *basis-function modeling approach* (see Section 3.1) used by [2] and [3] represents a bottom-up approach that uses a lower level encoding of a musical score in order to learn how different aspects of the score contribute to generate an expressive performance of a musical piece.

Grachten and Krebs [26], and Van Herwaarden et al. [27] present an alternative, unsupervised approach to modeling musical dynamics using restricted Boltzmann machines. This approach uses a piano roll representation of musical scores to explain the musical dynamics of performed piano music. In order to predict expressive dynamics of a score, the features learned by this model are trained in a supervised fashion using *least squares* regression. The choice of a note-centered representation of a musical score makes this system able to model harmonic context based on relative pitch, but insensitive to absolute pitch. Furthermore, this encoding of a score does not include performance directives written by the composer, such as dynamics or articulation markings (such as *piano*, staccato, etc). Both the KTH system and prior work on basis-function modeling have shown that the encoding of pitch and dynamics/articulation markings plays an important role in the rendering of expressive performances.

To date, there are (to the best of our knowledge) no computational models of ensemble performance in the sense that we described in Section 1 above. A slightly related method is described by [28]. They train a model on piano duets, with the aim of predictive modeling of musical expression in order to perform automatic musical accompaniment of a human performer.

3 A computational expression model for ensemble performance

In this Section, we describe the core contribution of this paper, a computational model that predicts the intensity of a recorded ensemble performance over time, as a function of the musical score. We begin by introducing the basis-function modeling approach (Section 3.1), that has been used before in a solo instrument setting. Next, we present three variants of the basis-modeling approach: a simple linear model, and two non-linear, neural network models (Section 3.2). Finally, we discuss how the ensemble setting is different from the solo instrument setting, and how the basis-function modeling approach is extended to deal with ensemble performances (Section 3.3).

3.1 Basis-function representations of musical information

In this section, we describe the basis-function modeling (BM) approach described by [2]. In this approach, a *musical score* is regarded as a set of elements on a time axis. This set includes note elements (with attributes like pitch, duration, metrical position) as well as non-note elements (e.g. dynamics and articulation markings). The set of all note elements in a score is denoted by $\mathcal{X}$. Musical scores can be described in terms of *basis-functions*, i.e. numeric descriptors that represent aspects of the score. Formally, we can define a basis-function φ as a real valued mapping $\varphi\colon \mathcal{X} \mapsto \mathbb{R}$. In a similar way, musical expression is characterized in a quantitative way by a number of *expressive parameters*. In particular, expressive dynamics can be conveyed by the MIDI velocities of notes performed on an appropriate device such as an electronic piano or a piano equipped with sensors. Further expressive parameters capture aspects of note timing and local tempo (e.g. inter-onset intervals between consecutive notes), and articulation (the proportion of the duration of a note with respect to its inter-onset interval). Although the basis-function approach can be applied without any alteration to model all of these expressive parameters, the focus in this study will be on expressive dynamics.

By defining basis-functions as functions of notes, instead of functions of time, the BM framework allows for modeling forms of music expression related to simultaneity of musical events, like the micro-timing deviations of note onsets in a chord, or the melody lead [29] in piano performance, i.e. the accentuation of the melody voice with respect to the accompanying voices by playing it louder and slightly earlier. However, expressive information for individual notes is difficult to obtain, and in situations where this information is not available (as in the present study), we represent expressive information as a function of time, rather than a function of notes. We return to this issue in Section 3.3.

Figure 1 illustrates the idea of modeling expressive dynamics using basis-functions schematically. Although basis-functions can be used to represent arbitrary properties of the musical score (see Section 3.1), the BM framework was proposed with the specific aim of modeling the effect of *dynamics markings*. Such markings are hints in the musical score, to play a passage with a particular dynamical character. For example, a *p* (for *piano*) tells the performer to play a particular passage softly, whereas a passage marked *f* (for *forte*) should be performed loudly. Such markings, which specify a constant loudness that lasts until another such directive occurs, are modeled using a step-like function, as shown in the figure. A gradual increase/decrease of loudness (*crescendo*/*diminuendo*) is indicated by right/left-oriented wedges, respectively. Such markings are encoded by ramp-like functions. A third class of dynamics markings, such as *marcato* (i.e. the "hat" sign over a note), or textual markings like *sforzato* (*sfz*), or *forte piano* (*fp*), indicate the accentuation that note (or chord). This class of markings is represented through (translated) unit impulse functions. In the BM approach, the expressive dynamics are modeled as a combination of the basis-functions, as displayed in the figure.

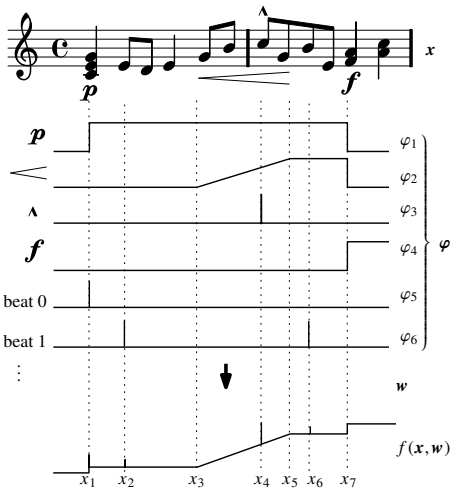


Figure 1. Schematic view of expressive dynamics as a function $f(\boldsymbol{x}, \boldsymbol{w})$ of basis-functions φ, representing dynamic annotations and metrical basis functions.

Groups of basis-functions As stated above, the BM approach encodes a musical score into a set of numeric descriptors. In the following, we describe various groups of basis-functions, each group representing a different aspect of the score. This list should by no means be taken as an exhaustive (or accurate) set of features for modeling musical expression. It is a tentative list that encodes basic information, either directly available, or easily computable from a symbolic representation of the musical piece (such as MusicXML[1]).

1. **Dynamics markings**. Bases that encode dynamics markings, such as shown in Figure 1. For each of the constant loudness markings (*p*, *pp*, *f* etc.), two additional ramp-function are included that allow for a gradual change towards the loudness level indicated by the marking. Such bases are referred to as *anticipation* functions, and we distinguish between *long* and *short* anticipations, according to how gradual is the change towards the target dynamics marking. Additionally, basis-functions that describe gradual changes in loudness, such as *crescendo* and *diminuendo*, are represented through a combination of a ramp function, followed by a constant (step) function, that continues until a new constant dynamics marking (e.g. *f*) appears, as illustrated by φ_2 in Figure 1.
2. **Polynomial pitch model**. Grachten and Widmer [2] proposed a third order polynomial model to describe the dependency of dynamics on pitch. This model can be integrated in the BM approach by defining each term in the polynomial as a separate basis-function, i.e. "pitch", "pitch^2", and "pitch^3". For transposing instruments, such as some of the wind instrument found in orchestras, the actual sounding pitch (concert pitch) is used.

[1] `http://www.musicxml.com`

3. **Vertical neighbors**. Three basis-functions that evaluate to the number of simultaneous notes with lower pitches, higher pitches, or the sum of both, respectively.
4. **Duration**. A basis-function that encodes the duration of a note.
5. **Metrical**. Representation of the time signature of a piece, and the position of each note in the bar. For example, the basis-function labeled *4/4 beat 0* evaluates to 1 for all notes that start on the first beat in a 4/4 time signature, and to 0 otherwise. This is illustriated by φ_5 and φ_6 in Figure 1 for the first and second beat in each bar.
6. **Accent**. Accents of individual notes or chords, such as the *marcato* in Figure 1.
7. **Staccato**. Encodes *staccato* markings on a note, an articulation indicating that a note should be temporally isolated from its successor, by shortening its duration

3.2 Linear vs. non-linear modeling

Basis-function modeling provides a way of representing diverse aspects of score information in a uniform way. The next question is how this information is used to model expressive parameters. Initial versions of the basis-function expression model used a linear model [30]. In a linear model, the expressive parameters are simply a weighted sum of the basis-functions, where the parameters of the model are the weights for each basis-function, to be estimated based on training data. A major advantage of a linear model is that the link between the basis-functions and the predictions is very clear: the weight for a basis-function expresses how strong the basis-function influences the output. This makes it easy to perform a qualitative analysis of what the model has learned, and by fitting the model on a particular piece, or on several pieces by the same performer, the weights may also capture characteristics of the expressive quality of a piece, or a performer. See [31] for an example of this.

The simplicity of linear modeling is at the same time a drawback. There are two main limitations to the linear approach. Firstly, the shape of a basis-function can only be used literally (apart from scaling and vertical translation) to approximate an expressive parameter. For example, a *crescendo* annotation is schematically represented as a *ramp function*, and this means that any increase of loudness in that region can only be approximated as a linear slope. In reality, it is likely that the shape of the loudness increase is not strictly linear. Secondly, the linear approach does not model any interactions between basis-functions.

To overcome these limitations, Cancino and Grachten [3] proposed a non-linear basis-function model for expression, based on a *feed-forward neural network*, where they ran experiments on Chopin piano music. The Discussion Section of that paper provides an example that shows the benefit of the non-linearity of the model, both in the non-linear transformation of the basis-functions, and in the interaction between basis-functions. More specifically, the example shows that the non-linear model reduces the effect of a crescendo in situations where the crescendo sign is directly preceded by a diminuendo sign. Such interactions

are not possible in a linear model. The example also shows that the ramp shape of the crescendo is slightly smoothed. We refer to the paper for more details.

A more powerful type of non-linear modeling can be obtained by introducing recurrence relations to the neural network architecture: *Recurrent Neural Networks* (RNNs) are a particular kind of discrete–time dynamical artificial neural networks (ANNs) suited for analyzing sequential data, such as time-series. These dynamic models have been successfully used for generating text sequences, handwriting synthesis and modeling motion capture data [32]. The structure of an RNN is similar to that of a feed forward neural network, with the particularity that it allows connections among its states associated with time delays. It is through these connections that RNNs are able to capture temporal correlations between events [33].

3.3 From solo piano to orchestral ensembles

The basis-function modeling approach described above has been developed for the purpose of modeling expression in solo piano performances, based on precise measurements obtained from a computer-controlled grand piano. There are several issues to be dealt with in order to apply the same approach to orchestral performances. In the rest of this Section, we will discuss these issues, and provide solutions.

Measured versus computed expressive parameters In a piano, the degrees of freedom for sound production, and therefore expressive performance, are limited to only a few, well-defined dimensions (such as hammer velocity, timing of key press and release) that can be measured relatively easily. Through the use of computer-controlled pianos [34], it is possible to obtain precise measurements of these dimensions in piano performances. Similar measurements are typically not easily possible for other classes of instruments, such as bowed string instruments, or wind instruments, which have more complex sound production mechanisms. Although with the appropriate sensors, rich descriptions of non-piano performances may be obtained (for instance to measure the bending of the reed in wind instruments [35] or bow movements in violin playing [36]), the usage of such sensors is often intrusive, and thus limited to experimental setups. Moreover, data recorded in this way is prone to noise, and bulky in case of large ensembles.

For these reasons, our current work is focused on relatively coarse, but easy to obtain form of expressive information, namely the instantaneous overall loudness computed from an audio recording of a professional music performance. This implies that there is only a single value for each expressive parameter at each *time instant*, as opposed to the measured piano scenario, where expressive parameters can be defined in part for *individual notes*, even if they occur at the same time instant. Since the basis-functions of the form described in Section 3.1 return a value for each note, and thus possibly multiple values for a single time instant, it is necessary to fuse these values in order to obtain a single prediction for the expressive parameter at that time instant.

Indexing basis-functions The basis-modeling approach, including the list of basis-functions defined in Section 3.1, is designed to generate a set of basis-functions, given a *score part* for an instrument. When training an expression model on a data set containing performances of multiple pieces, the basis-functions produced for each piece must be mapped to each other. In the solo instrument setting, this mapping is done on the basis of labels that are uniquely assigned to each basis-function. In orchestral scores, the labels are not unique any longer, since the same basis-functions are produced for each instrument (coding the same type of score information, but for different instruments). To deal with this, it is necessary to index the basis-functions by the tuple *(instrument name, basis-function label)*.

A further issue is that the notated instrument names in the score, do not follow any strict standard. Instrument names may be written in different languages (e.g. "Fagott", "bassoon"), and may be abbreviated (e.g. "Vln." for violin, "Cl." for clarinet). To overcome this issue, the instrument names and abbreviations extracted from the score are matched to one of a set of *canonical instrument names* (the unabbreviated English names), using string matching techniques.

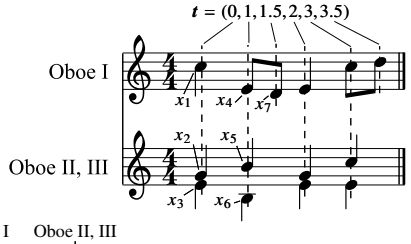
Merging and fusion of basis-functions within instrument classes In orchestral scores, there may be several instances (voices) of an instrument, usually designated by numbers (e.g. "Violin 1", "Violin 2"). Furthermore, multiple instances of an instrument may share a single staff.

The occurrence of multiple instruments of the same type poses a problem for training the model, since it is not clear how the mapping of basis-functions across pieces should be defined in order to create a consistent dataset consisting of multiple pieces. For instance, when one piece involves a single violin, and another piece involves two violins, the question which of the two violins in the latter piece should be mapped to the violin in the first piece is arbitrary, and moreover, it is unclear how to deal with the remaining, unmapped violin.

For this reason, we choose to combine all instances of the same instrument class into a single set of basis-functions, using a *fusion operation* that can be specified per basis-function type. In this way, for each piece there is a single set of basis-functions conveying the activity of a given instrument *class*, rather than one set for each *instance* of that class.

The process of merging and fusion is shown in Figure 2. First, a collection of K predefined basis-functions $\boldsymbol{\Phi} = (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_K)$ is applied to each score part, where e.g. $\varphi_1 := \varphi_{\text{pitch}}$ corresponds to the pitch of a note, expressed as the MIDI note number. This gives a matrix of basis-function values for each score part. Notes occurring at the same time are laid out consecutively, as shown in the leftmost matrix in Figure 2; Note that the two score parts' matrices were already stacked together here, with the rows arranged according to the notes' onset times.

Second, the score information of different score parts belonging to the same instrument class, here "Oboe", need to be combined into a single instrument class matrix. This is referred to as *merging*. As can be seen from the matrix in



		Oboe I		Oboe II, III	
t	note	φ_{pitch}	φ_{dur}	φ_{pitch}	φ_{dur}
0	x_1	72	1	-	-
0	x_2	-	-	67	1
0	x_3	-	-	64	1
1	x_4	64	0.5	-	-
1	x_5	-	-	71	1
1	x_6	-	-	59	1
1.5	x_7	62	0.5	-	-

t	note	φ_{pitch}	φ_{dur}
0	x_1	72	1
0	x_2	67	1
0	x_3	64	1
1	x_4	64	0.5
1	x_5	71	1
1	x_6	59	1
1.5	x_7	62	0.5

t	note	φ_{pitch}	φ_{dur}
0	x'_1	68	1
1	x'_2	65	0.83
1.5	x'_3	62	0.5

Figure 2. Illustration of *merging* and *fusion* of score information of two different parts belonging to the same instrument class "Oboe". The matrix on the left shows the basis-functions φ_{pitch} and φ_{dur} for each of the two score parts (truncated after the first few notes). Note the consecutive layout of simultaneously occurring notes, as indicated by the first column of each matrix, giving the notes' onset times. The matrix in the center illustrates *merging*, where the data of both score parts were combined. Finally, the third matrix on the right is the result of *fusion* operations, applied per basis function to each set of values occurring at the same time point. See text for further explanation.

the center, the corresponding columns of the leftmost matrix were stacked into one column each, with simultaneous notes still consecutively listed. The number of columns is the cardinality of the set of the basis-functions of the involved score parts.

Finally, a fusion operation is applied to each subset of a column having the same onset time t, yielding a single value φ for each time instant. The matrix on the right in Figure 2 results from applying fusion to the matrix in the center. The number of rows is given by the size of the union of all occurring onset times. Following this procedure, for each instrument class in a piece, there will be a single collection of basis-functions that can easily be mapped to other pieces' basis function of the same instrument class. Thereby, a collection of matrices $\boldsymbol{\Phi}_i$ for the instrument classes $i = 1, 2, \ldots, I$ of a piece is produced.

Aggregation of basis-functions of instrument classes in a piece After collecting per instrument class basis-function matrices, we need one final step to conclude the data extraction for a piece $\mathcal{P}$. All instrument classes' data are aggregated into a single per-piece matrix $\boldsymbol{\Phi}_{\mathcal{P}}$. The number of rows of this matrix is given by the total number of unique onset times across all score parts P and is denoted $N_{\mathcal{P}}$. The number of columns of $\boldsymbol{\Phi}_{\mathcal{P}}$ is given by $K_{\mathcal{P}}$, the sum of the number of columns of the single instrument class matrices $\boldsymbol{\Phi}_i$, thus $\boldsymbol{\Phi}_{\mathcal{P}} \in \mathbb{R}^{N_{\mathcal{P}} \times K_{\mathcal{P}}}$.

Model description For the training procedure across multiple pieces, it is necessary to match all the pieces' basis-functions to each other. This can again be achieved by appropriately stacking together all involved $\boldsymbol{\Phi}_{\mathcal{P}}$ to produce a data set matrix $\boldsymbol{\Phi}_{\mathcal{S}}$ of shape $(N_{\mathcal{S}} \times K_{\mathcal{S}})$. $N_{\mathcal{S}}$ is the sum of the number of rows of all per-piece matrices $\boldsymbol{\Phi}_{\mathcal{P}}$, whereas $K_{\mathcal{S}}$ is the cardinality of the set of all uniquely occurring basis functions across the data set.

The model can now be described in the following way. In general, an expressive target parameter $\boldsymbol{y}$ is modeled as a function $f(\cdot)$ of the data $\boldsymbol{\Phi}_{\mathcal{S}}$ extracted from the scores and a vector $\boldsymbol{w}$ of weights:

$$\boldsymbol{y} = f(\boldsymbol{\Phi}_{\mathcal{S}}, \boldsymbol{w}) + \boldsymbol{\varepsilon} \tag{1}$$

Here, $\boldsymbol{w}$ has shape $(K_{\mathcal{S}} \times 1)$, and $\boldsymbol{\varepsilon}$ is zero mean Gaussian noise with covariance matrix $\sigma^2 \boldsymbol{I}$, with $\boldsymbol{I}$ an identity matrix of appropriate shape.

In a linear model, the function $f(\cdot)$ is a linear combination:

$$\boldsymbol{y} = \boldsymbol{\Phi}_{\mathcal{S}}\, \boldsymbol{w} + \boldsymbol{\varepsilon}. \tag{2}$$

In the following Section 4, we describe the experiments that used the linear model, a Feed Forward Neural Network and a Recurrent Neural Network (RNN) for estimating the model parameters (weights) $\hat{\boldsymbol{w}}$. Once the estimated parameters are established, they can be used to predict the loudness variations $\hat{\boldsymbol{y}}_{\mathcal{P}}$ from the score information $\boldsymbol{\Phi}_{\mathcal{P}}$ of a piece (not yet seen by the model):

$$\hat{\boldsymbol{y}}_{\mathcal{P}} = f(\boldsymbol{\Phi}_{\mathcal{P}}, \hat{\boldsymbol{w}}). \tag{3}$$

We will not go into more details about the model here but instead refer the interested reader to Cancino and Grachten [3] for a more detailed and formal explanation.

4 Experiments

In our experiment, we want to assess how well we can predict variations in loudness (as an expressive parameter) of ensemble pieces using the basis-function model. We compare different models, namely a linear model, a Feed Forward Neural Network (FFNN) and a Recurrent Neural Network (RNN).

4.1 Data

The corpus used for the experiments is summarized in Table 1 below. It consists of symphonies from the classic and romantic period.

For each of these symphonies, a recorded performance (an audio file), a machine-readable representation of the musical score (a MusicXML file) and an automatically produced, manually corrected alignment between score and performance are available in the corpus.

Table 1. Pieces used in the experiments.

Composer	Piece	Cond.	Movements
Beethoven	S. 6 in F-Maj. (op. 68)	Fischer	1, 2, 3, 4, 5
Beethoven	S. 9 in D-Min. (op. 125)	Fischer	1, 2, 3, 4
Mahler	S. 4 in G-Maj.	Jansons	1, 2, 3, 4
Bruckner	S. 9 in D-Min. (WAB 109)	Jansons	1, 2, 3

We used recordings of performances by the Royal Concertgebouw Orchestra conducted by Ivan Fischer or Mariss Jansons, all performed at the Royal Concertgebouw in Amsterdam, the Netherlands. Since the various movements of a symphony are handled individually, from now on each movement is referred to as a piece. The corpus thus amounts to a total of 16 pieces. The corresponding performances sum up to a total length of almost 4 hours of music. From the 16 pieces' scores, a total of $N_{\mathcal{S}} = 47228$ note onsets, belonging to $K_{\mathcal{S}} = 1518$ basis functions, were extracted.

The symbolic scores used for the extraction of the basis-functions were provided partly by Bärenreiter Verlag[2], and partly by Donemus Publishing[3]. The target values (loudness) for each piece were extracted using the loudness measure defined by EBU R128, as described in [37].

The recordings were all made in the same hall and produced for the same target medium. Thus, we do not expect the recording and production process to be a significant source of variation in loudness.

To map the note onset times in the score–the positions at which basis functions are evaluated–to loudness values in the recorded performance, the score-to-audio alignment technology as described in [38] was used. The alignments were corrected by a human annotator at least at the level of single bars. It makes sense to estimate the loudness corresponding to a particular score note by measuring the loudness slightly *after* the estimated onset time in the performance. One reason for this is that some instruments have a significant attack-time, meaning that peak loudness occurs some time after the start of the note. Another reason is that in the possible case of some minor residual error in the alignment after correction, the probability that a particular note estimated to start at t is actually sounding is higher at $t+\delta$ than at t for some positive δ, assuming sum of the average residual alignment error and the chosen δ is smaller than the average note duration. For these reasons, we extracted the target value 1/10th of a beat after the onset time point given by the alignment to decrease the probability of "hitting" a note before its onset.

4.2 Method

We used a leave-one-out scenario where the model is trained on 15 of the 16 pieces and then is used to predict the target values for the unseen remaining

[2] `http://www.baerenreiter.com`

[3] `http://www.donemus.nl`

piece. The non-linear models (FFNN, and RNN) are trained by gradient descent optimization. Both the feed-forward and the recurrent neural network each were set up with a single hidden layer of 20 units. From the 15 training pieces, two pieces were kept for validation, to avoid overfitting the models to the training data, a practice known as *early stopping* [39]. The predictions are evaluated with respect to the target (here loudness curve) in terms of the Pearson correlation coefficient r and the coefficient of determination R^2.

The set of basis-functions used in the experiments encode note pitch, duration, and metrical position, the number of simultaneous notes within instrument groups, inter-onset intervals between subsequent notes, repeat signs, note accent, staccato, fermata signs, and dynamics markings. A full description of the basis functions is omitted for brevity.

4.3 Results and discussion

The results of the experiments are shown in the following Table 2. For each piece, we report the *Coefficient of determination* (R^2) that measures the proportion of variance in the recorded loudness curve that is explained by the model, and *Pearson's correlation coefficient* (r), that measures the strength of the linear dependence between the recorded and the predicted loudness curves. The R^2 measure has an upper bound of 1, and has no lower bound (predictions can be arbitrarily far away from the target values). Positive R^2 values indicate that the models perform better than the baseline of predicting the mean value of the loudness over the whole piece.

Several observations can be made from Table 2. First of all, both the R^2 and the r values for Lin are generally lower than those for FFNN and RNN, demonstrating that the non-linear modeling provides a clear advantage over the linear modeling approach. Given the relatively small data set, this is surprising, since the FFNN and RNN have much more parameters than the Lin model, and are therefore more prone to both *overfitting* or *underfitting*. Secondly, the RNN model provides more accurate predictions than the FFNN model, although this advantage is less prominent than the advantage over the linear model.

A possible explanation for the advantage of the RNN model lies in a limitation of the basis-function modeling approach in its current form. The example in Figure 2 illustrates this limitation. Notes x_5 and x_6, starting at beat 1 are quarter notes, still sounding at the onset of note x_7 at beat 1.5. Thus, it is to be expected that the presence of notes x_5 and x_6 will affect the overall loudness value at beat 1.5. However, the basis-functions representing those notes are only active at beat 1, and not at beat 1.5 (the only row pertaining to beat 1.5 in the left matrix is the one representing x_7). The linear model and the FFNN have no way to incorporate basis-function information describing notes x_5 and x_6 at time 1.5, but through its recurrent connections, the RNN *can* learn that information at prior time steps can be helpful to predict the loudness at the current time step. To verify this explanation, a further analysis of the results is necessary, which is beyond the scope of this paper.

Table 2. Predictive accuracy in a leave-one-out scenario for different models; MSE = mean squared error (smaller is better); R^2 = coefficient of determination (larger is better); r = Pearson correlation coefficient (larger is better); RN = recurrent neural network; FF = feed forward neural network; Lin = linear model; Best value per piece and measure emphasized in bold

Piece		MSE			R^2			r		
		RN	FF	Lin	RN	FF	Lin	RN	FF	Lin
LvB S6	Mv 1	**0.57**	0.60	0.96	**0.43**	0.40	0.04	**0.71**	0.66	0.40
	Mv 2	**0.80**	0.87	0.94	**0.20**	0.13	0.06	**0.45**	0.36	0.35
	Mv 3	**0.40**	0.45	0.56	**0.60**	0.55	0.44	**0.79**	0.76	0.67
	Mv 4	**0.66**	0.67	0.87	**0.34**	0.33	0.13	**0.61**	0.60	0.42
	Mv 5	**0.52**	0.59	0.66	**0.48**	0.41	0.34	**0.74**	0.68	0.58
Mah S4	Mv 1	**0.64**	0.76	6.21	**0.36**	0.24	-5.21	**0.60**	0.51	0.02
	Mv 2	**0.95**	0.98	11.69	**0.05**	0.02	-10.69	**0.26**	0.22	0.03
	Mv 3	**0.51**	0.66	2.63	**0.49**	0.34	-1.63	**0.71**	0.59	0.19
	Mv 4	**0.86**	0.96	2.03	**0.14**	0.04	-1.03	**0.40**	0.29	0.18
LvB S9	Mv 1	**0.62**	0.67	0.70	**0.38**	0.33	0.30	**0.63**	0.60	0.58
	Mv 2	**0.44**	0.57	0.67	**0.56**	0.43	0.33	**0.75**	0.66	0.59
	Mv 3	**0.95**	0.98	1.24	**0.05**	0.02	-0.24	0.27	**0.29**	0.26
	Mv 4	**0.62**	0.80	0.92	**0.38**	0.20	0.08	**0.63**	0.55	0.45
Bru S9	Mv 1	**0.41**	0.52	7.22	**0.59**	0.48	-6.22	**0.77**	0.70	0.24
	Mv 2	**0.39**	0.48	0.97	**0.61**	0.52	0.03	**0.80**	0.74	0.47
	Mv 3	**0.61**	0.65	0.99	**0.39**	0.35	0.01	**0.65**	0.59	0.33

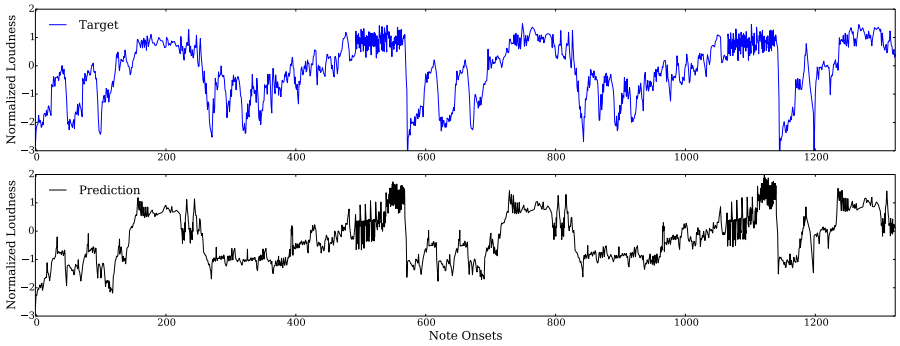


Figure 3. Prediction of loudness variation in movement 3 of Beethoven's Symphony No. 6. The upper curve shows the (normalized) loudness extracted from the audio recording, the lower curve the loudness variations predicted by the recurrent neural network (RNN), based on the written score.

Furthermore, Table 2 shows that all models have difficulty predicting the loudness curves for some of the pieces, in particular for the 2nd movement of Mahler's 4th Symphony and the 3rd movement of Beethoven's Symphony No. 9. Since the data set is relatively small, we hypothesized that the inaccurate predictions for these pieces might be due to the occurrence of *singular* basis-functions: basis-functions that are active in the test piece, but not (or hardly) active in any training piece. This may result in an undertraining of the models for these basis-functions, leading to inaccurate predictions of the loudness curves. However, upon testing this, we found that the pieces with low predictive accuracy did not have substantially larger numbers of singular basis-functions than other pieces. For further investigation a sensitivity analysis may be helpful, to test whether the models are more sensitive to singular basis-functions for the problematic pieces than for other pieces. Should this be the case, the models may benefit from stronger regularization of the model parameters during training.

Figure 3 shows an example of a loudness curve extracted from a recorded performance, and the predicted loudness curve by the RNN (trained on other pieces), based on the written score. Note that although the details of the predicted loudness curve are not very accurate, the overall shape of the predicted curve resembles the actual loudness curve.

5 Conclusion and future work

In this paper, we have described an extension of an existing model for musical expression for solo piano to deal with performances of ensembles, such as a symphonic orchestra. The model represents score information for each instrument in the ensemble, and uses the joint information to predict the overall loudness curve of a recorded performance. We have evaluated three variants of the model

(one linear version and two non-linear versions), on a dataset of recorded performances of symphonic music pieces by Mahler, Beethoven, and Bruckner, played by the Royal Concertgebouw Orchestra. Although the data set is rather small, the experiments show that the non-linear models have a clear advantage over the linear model, and also that the recurrent non-linear model performs better than the feed forward non-linear model.

Arguably the overall loudness curve of a performance is a very coarse way of representing expressive variation of dynamics. For more precise and reliable modeling, it is desirable to have loudness values available per instrument, or per instrument class. A set of possibly useful recordings (where each instrument of the orchestra is recorded in isolation) is reported in [40]. For pragmatic and technical reasons however, it is unfeasible to record each instrument separately in live performances.

Source separation techniques such as reported in [41], and [42], may also be useful for more precise expression modeling, as they provide a means to separate instrument families from an orchestral recording, to be used for modeling loudness of individual instrument sections.

Acknowledgements. This work is supported by the European Union's Seventh Framework Programme FP7 / 2007-2013 (projects PHENICX / grant number 601166 and Lrn2Cre8 / grant number 610859), and by the European Research Council (ERC) under the EU's Horizon 2020 Framework Programme (ERC Grant Agreement number 670035, project CON ESPRESSIONE). Furthermore, we wish to thank the Royal Concertgebouw Orchestra, in particular Marcel van Tilburg and David Bazen, for providing the audio recordings used in this study.

References

1. Andreas Arzt, Gerhard Widmer, and Simon Dixon. Automatic page turning for musicians via real-time machine listening. In *ECAI*, pages 241–245, 2008.
2. M. Grachten and G. Widmer. Linear basis models for prediction and analysis of musical expression. *Journal of New Music Research*, 41(4):311–322, 2012.
3. C. E. Cancino Chacón and M. Grachten. An evaluation of score descriptors combined with non-linear models of expressive dynamics in music. In Nathalie Japkowicz and Stan Matwin, editors, *Proceedings of the 18th International Conference on Discovery Science (DS 2015)*, Lecture Notes in Artificial Intelligence, Banff, Canada, 2015. Springer.
4. M. Grachten and C. E. Cancino Chacón. Temporal dependencies in the expressive timing of classical piano performances. 2016. Submitted.
5. A. Binet and J. Courtier. Recherches graphiques sur la musique. L'année Psychologique (2), 201–222, 1896.
6. C. E. Seashore. *Psychology of Music*. McGraw-Hill, New York, 1938. (Reprinted 1967 by Dover Publications New York).
7. Patrik N. Juslin. Emotional communication in music performance: A functionalist perspective and some data. *Music Perception: An Interdisciplinary Journal*, 14(4):383–418, 1997.

8. P. Juslin and J. Sloboda. *Music and Emotion: Theory and Research*. Oxford University Press, 2001.
9. S. Canazza, G. De Poli, A. Rodá, and A Vidolin. An abstract control space for communication of sensory expressive intentions in music performance. *Journal of New Music Research*, 32(3):281–294, 2003.
10. E. F. Clarke. Expression and communication in musical performance. In Johan Sundberg, Lennart Nord, and Rolf Carlson, editors, *Music, Language, Speech and Brain*. MacMillan Academic and Professional Ltd, 1991.
11. N.P. Todd. A computational model of rubato. *Contemporary Music Review*, 3 (1), 1989.
12. J. A. Sloboda. The communication of musical metre in piano performance. *Quarterly Journal of Experimental Psychology*, 35A:377–396, 1983.
13. P.E. Keller. Ensemble performance: Interpersonal alignment of musical expression. In D. Fabian, R. Timmers, and E. Schubert, editors, *Expressiveness in music performance: Empirical approaches across styles and cultures*, pages 260–282. Oxford University Press, 2014.
14. Werner Goebl and Caroline Palmer. Synchronization of timing and motion among performing musicians. *Music Perception*, 26(5):427–438, 2009.
15. Laura Bishop and Werner Goebl. Context-Specific Effects of Musical Expertise on Audiovisual Integration. *Frontiers in Cognitive Science*, pages 1–24, September 2014.
16. Donald Glowinski, Arianna Riolfo, Kanika Shirole, Kim Torres-Eliard, Carlo Chiorri, and Didier Grandjean. Is he playing solo or within an ensemble? how the context, visual information, and expertise may impact upon the perception of musical expressivity. *Perception*, 43(8):825–828, 2014.
17. Gerhard Widmer and Werner Goebl. Computational models of expressive music performance: The state of the art. *Journal of New Music Research*, 33(3):203–216, 2004.
18. G. Widmer. Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research*, 31(1):37–50, 2002.
19. W. L. Windsor and E. F. Clarke. Expressive timing and dynamics in real and artificial musical performances: using an algorithm as an analytical tool. *Music Perception*, 15(2):127–152, 1997.
20. K. Teramura, H. Okuma, Y. Taniguchi, S. Makimoto, and Maeda S. Gaussian process regression for rendering music performance. In *Proceedings of the 10th International Conference on Music Perception and Cognition (ICMPC)*, Sapporo, Japan, 2008.
21. G. Grindlay and D. Helmbold. Modeling, analyzing, and synthesizing expressive piano performance with graphical models. *Machine Learning*, 65(2–3):361–387, 2006.
22. G. De Poli, Canazza S., A Rodà, A. Vidolin, and P. Zanon. Analysis and modeling of expressive intentions in music performance. In *Proceedings of the International Workshop on Human Supervision and Control in Engineering and Music*, Kassel, Germany, September 21–24 2001.
23. G. Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146(2):129–148, 2003.
24. A. Friberg, R. Bresin, and J. Sundberg. Overview of the kth rule system for musical performance. *Advances in Cognitive Psychology*, 2(2–3):145–161, 2006.
25. R. Bresin. Artificial neural networks based models for automatic performance of musical scores. *Journal of New Music Research*, 27 (3):239–270, 1998.

26. M. Grachten and F. Krebs. An assessment of learned score features for modeling expressive dynamics in music. *IEEE Transactions on Multimedia*, 16(5):1211–1218, 2014.
27. Sam van Herwaarden, Maarten Grachten, and W B de Haas. Predicting Expressive Dynamics using Neural Networks. In *Proceedings of the 15th Conference of the International Society for Music Information Retrieval*, pages 47–52, July 2014.
28. Guangyu Xia, Yun Wang, Roger B. Dannenberg, and Geoffrey Gordon. Spectral learning for expressive interactive ensemble music performance. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2015.
29. W. Goebl. Melody lead in piano performance: expressive device or artifact? *Journal of the Acoustical Society of America*, 110(1):563–572, 2001.
30. M. Grachten and G. Widmer. Explaining expressive dynamics as a mixture of basis functions. In *Proceedings of the Eighth Sound and Music Computing Conference (SMC)*, Padua, Italy, 2011.
31. M. Grachten and G. Widmer. A method to determine the contribution of annotated performance directives in music performances. In *Proceedings of the International Symposium of Performance Science*, Toronto, Canada, 2011.
32. Alex Graves. Generating Sequences With Recurrent Neural Networks. *arXiv*, 1308:850, 2013.
33. Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to Construct Deep Recurrent Neural Networks. *arXiv*, pages 1–13, April 2014.
34. R. A. Moog and T. L. Rhea. Evolution of the Keyboard Interface: The Bösendorfer 290 SE Recording Piano and the Moog Multiply-Touch-Sensitive Keyboards. *Computer Music Journal*, 14(2):52–60, 1990.
35. Alex Hofmann, Vasileios Chatziioannou, Michael Weilguni, Werner Goebl, and Wilfried Kausel. Measurement setup for articulatory transient differences in woodwind performance. In *Proceedings of Meetings on Acoustics*, volume 19, pages 035–060. Acoustical Society of America, 2013.
36. E. Schoonderwaldt and M. Demoucron. Extraction of bowing parameters from violin performance combining motion capture and sensors. 126(5):2695, 2009.
37. EBU-R-128. Bu tech 3341-2011, practical guidelines for production and implementation in accordance with ebu r 128, 2011.
38. Maarten Grachten, Martin Gasser, Andreas Arzt, and Gerhard Widmer. Automatic alignment of music performances with structural differences. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, November 2013.
39. N. Morgan and H. Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 630–637. Morgan-Kaufmann, 1990.
40. Jukka Pätynen, Ville Pulkki, and Tapio Lokki. Anechoic recording system for symphony orchestra. *Acta Acustica united with Acustica*, 94(6):856–865, 2008.
41. Ricard Marxer, Jordi Janer, and Jordi Bonada. Low-latency instrument separation in polyphonic audio using timbre models. In *Latent Variable Analysis and Signal Separation*, pages 314–321. Springer Berlin Heidelberg, 2012.
42. Marius Miron, Julio José Carabias-Orti, and Jordi Janer. Improving score-informed source separation for classical music through note refinement. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2015.