Preprint version; final version available at http://ieeexplore.ieee.org
RO-MAN (2015), pp: 175-721
Published by: IEEE

Towards a Synchronised Grammars Framework for Adaptive Musical Human-Robot Collaboration

Miguel Sarabia*, Kyuhwa Lee† and Yiannis Demiris*

Abstract— **We present an adaptive musical collaboration framework for interaction between a human and a robot. The aim of our work is to develop a system that receives feedback from the user in real time and learns the music progression style of the user over time. To tackle this problem, we represent a song as a hierarchically structured sequence of music primitives. By exploiting the sequential constraints of these primitives inferred from the structural information combined with user feedback, we show that a robot can play music in accordance with the user's anticipated actions. We use Stochastic Context-Free Grammars augmented with the knowledge of the learnt user's preferences.**

We provide synthetic experiments as well as a pilot study with a Baxter robot and a tangible music table. The synthetic results show the synchronisation and adaptivity features of our framework and the pilot study suggest these are applicable to create an effective musical collaboration experience.

I. Introduction

Everybody enjoys music. It is not surprising then that there is intense research in the field of human-robot musical collaboration [1], [2], [3], [4], [5], [6]. However there are many challenges to having a robot playing music alongside a human: there are small-scale problems like actuation speed and accuracy, as well as large-scale issues such as the progression of the musical pattern. There is also the question of adaptation, how can a robot adapt to each user's musical taste? We focus on tackling two of these problems: synchronisation with the musical pattern and adaptation to the user's musical taste.

To solve these problems we make use of Stochastic Context-Free Grammars (SCFG). SCFGs—also known in the literature as Probabilistic Context-Free Grammars—have found use in fields such as natural language processing [7], RNA analysis [8], password cracking [9], computer vision [10] and robotics [11]. The reason for their widespread adoption lies in the intuitiveness and simplicity of their formulation (which we will revise in Section III-A). SCFGs can account for the underlying structure of musical compositions by representing the music primitives as the base symbols of the grammar (terminals) and the musical structure as a set of hierarchical rules (productions).

That is the approach we follow. Using a tangible music interface (Reactable) we designed a human-robot collaboration system where a user and a robot create music together. In this set-up, the user is the conductor and is in charge of the melody while the robot (Baxter) produces the drums accompaniment. Working with Baxter ensures that our algorithms are robust enough to deal with unexpected errors by the robot.

By using the synchronised grammars framework—which we will present in Section III—the robot can predict the next most likely action of the user and act accordingly. Our framework works with probability distributions over all musical primitives (terminals). This is essential as Baxter draws its next action from the computed probability distribution rather than selecting the action with highest probability. Consequently, our system may choose an unexpected action, but we argue this adds a creativity aspect to the collaboration.

We ran synthetic experiments with our framework and found that the amount of negative feedback was significantly reduced, as presented in Section IV. The results from the pilot study in Section V confirm this, as most participants agreed that the robot was able to become a better accompanist as the trials progressed.

Both the code for our synchronised grammars framework as well as the music collaboration controllers and sound samples will be open-sourced and made available from our group's website[1].

II. Related Work

Our work is at the intersection of three distinct areas of research: human-robot collaboration, music generation and Stochastic Context-Free Grammars. In what follows we present the literature of each field relevant to this paper.

In [12], Fong et al. present the requirements for human-robot collaboration based on dialogue. Though not all of these apply to our work (as our robot does not speak), they do note the importance of robot adaptiveness. According to them "the robot has to be able to adapt to different operators and to adjust its behaviour as needed". Consequently, the framework we present in this paper has a module just to adapt to the preferences of the user (see III-C). Fong et al. point out that any collaborative robot should be able to follow or ignore human advice depending on the circumstances. Our framework also provides the means to implement this decision mechanism.

Cicconet et al. developed a robotic system for human-robot collaborative percussion generation [1]. Their focus is on understanding of social cues (in particular, visual cues) to anticipate the next action the user will perform. This approach is complementary to ours, as we focus on

* Personal Robotics Lab, Department of Electrical and Electronic Engineering, Imperial College London, United Kingdom. ✉ {miguel.sarabia, y.demiris}@imperial.ac.uk

† Chair in Non-invasive Brain-machine Interface Lab, École Polytechnique Fédérale de Lausanne, Switzerland. ✉ kyu.lee@epfl.ch

[1] http://imperial.ac.uk/PersonalRobotics

Preprint version; final version available at http://ieeexplore.ieee.org
RO-MAN (2015), pp: 175-721
Published by: IEEE

predicting the intentions of the user based on the *structure* of the task.

The work of Hoffman et al. [13] in human-robot collaboration for task assembly is relevant as well. The authors present a Markov process framework that can anticipate user actions. Their tests in a simulator with 27 subjects show that users much preferred a robot with anticipation capabilities. This finding is echoed in [14] where experiments with an actual robot and 16 participants show that users spent 85% less time idling when the robot could anticipate their actions in a collaborative assembly task. Similar to both these works our framework predicts the most likely user action to decide what to do next.

Another robotic drummer is described in [2]. In this case, the approach to synchronisation is made from a developmental point of view. Nico, a humanoid robot, learns to integrate its sensory inputs (visual, auditory and proprioceptive) to produce the appropriate drum-beats. This contrasts with our approach where all the information for Baxter comes from the structure of the task and the tangible table itself.

With respect to automated music generation, already in 1986 Ebcioğlu developed an expert system to generate harmony to music in the style of J.S. Bach's chorales [15].

More recently, there has been work in applying machine learning techniques to music generation. For instance, [16] describes an architecture based on Echo State Networks to capture the *groove* of drummers. The authors define *groove* as variations in timing and musical pattern. We recognise the importance of these variations for the music produced to sound *natural* and our framework can indeed produce musical pattern variations, but not yet timing variations.

There are also approaches to music generation using Context-Free Grammars [3], [4]. In particular, [3] describes a system to improvise a drum accompaniment which chooses the pattern to be played based on both the current context and previous history. Whilst our approach is similar in that we also derive the final action from a mixture of distributions, we explicitly consider a secondary (influencing) grammar to represent the actions of the other performer. Our approach has the advantage that it predicts the most likely next action of the collaborator rather than merely reacting to the observed actions.

From a theoretical stand-point, in this article we show how to synchronise two independent SCFG parser in real-time. Zhang et al. present an offline framework for recognition of complex temporal events using a SCFG parser extended with Allen's temporal logic [17]. Their framework is able to generate the grammars automatically and can detect multi-agent actions such as two people meeting in the street. This system is able to represent the relations between a user and its robot collaborator, though it does so from a global perspective—that is, there is only one grammar to represent the whole interaction. Such generated grammar would be both less intuitive and more computationally expensive to parse than our approach.

III. Synchronised Grammars Framework

In this section we describe our framework to synchronise two Stochastic Context-Free Grammars and to add adaptability to this synchronisation mechanism. First off, however we start with a quick overview of SCFG parsing.

A. Stochastic Context-Free Grammars overview

Context-Free Grammars were first introduced by [18]. We base our work on the parsing algorithm by Stolcke et al. [19] who extended Earley's top-down parser [20] to SCFGs. To summarise, a grammar is defined by:

$$\mathcal{G} = (\mathcal{N}, \mathcal{T}, \mathcal{S}, \mathcal{R}, \mathcal{P})$$

where $\mathcal{N}$ is the set of non-terminals, $\mathcal{T}$ is the set of terminals, $\mathcal{S}$ is the starting non-terminal, $\mathcal{R}$ is the set of rules of the form $X \rightarrow \lambda$ with $X \in \mathcal{N}$ and $\lambda \in (\mathcal{N} \cup \mathcal{T})^*$, and $\mathcal{P}$ is the set of rule probabilities, that is:

$$\mathcal{P} = \bigcup_{\forall r \in \mathcal{R}} P(r)$$

The role of the parser consists in generating states of the following form:

$$i: \quad {}_kX \rightarrow \lambda.\mu \quad [\alpha, \gamma]$$

where i is state-set indicator and denotes at which point this state was created, k denotes at which point this chain was first considered by the parser, X is a non-terminal, λ and μ are a combination of terminals and non-terminals (and both could be empty), the dot represents the next symbol to be scanned, α is the forward probability (i.e. the probability of the parser generating this rule), γ is the inner probability (that is, the probability of generating the current string starting at point k).

There are three functions to generate all the required states that the parser will execute iteratively: *scan*, *complete* and *predict*. We leave it to the interested reader to check the details of each function [10], [19]. Suffice to say that *scan* is in charge of incorporating inputs (terminals) into the parser, *complete* takes care of moving the dots of non-terminals whose rules are finished and *predict* adds states to expand every non-finished non-terminal.

B. Prediction of next input

In order to perform synchronisation between two parsers it is vital that we are able to predict the next most likely step of each parser. Fortunately, the given formulation of Stochastic Context-Free Grammars allows us to do so easily.

Let us denote the state-set $\mathcal{SS}_i$ as the set of states introduced in step i. Importantly, $\mathcal{SS}_i$ are the only steps that *scan* will consider when incorporating the new input terminals into the parser.

More accurately, since *scan* only considers terminals, the only relevant states will have $\mu = x\nu$ with $x \in \mathcal{T}$ and $\nu \in (\mathcal{N} \cup \mathcal{T})^*$. We define this set of states as the candidate set:

$$\mathcal{CS}_i = \bigcup_{\forall x \in \mathcal{T}} (i: {}_kX \rightarrow \lambda.x\nu\,[\alpha, \gamma])$$

Preprint version; final version available at http://ieeexplore.ieee.org
RO-MAN (2015), pp: 175-721
Published by: IEEE

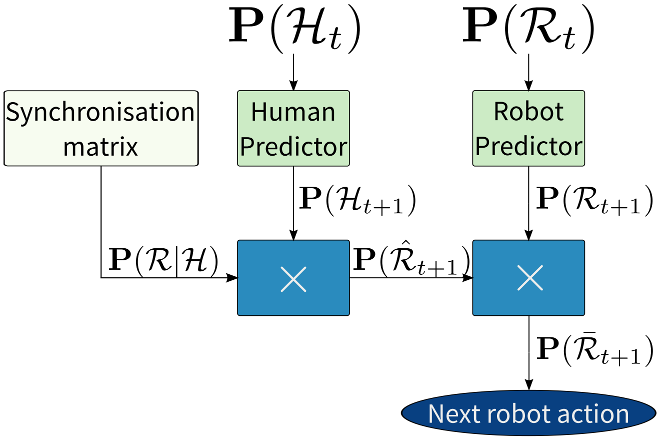
$\mathbf{P}(\mathcal{H}_t)$
$\mathbf{P}(\mathcal{R}_t)$
Synchronisation matrix
Human Predictor
Robot Predictor
$\mathbf{P}(\mathcal{H}_{t+1})$
$\mathbf{P}(\mathcal{R}_{t+1})$
$\mathbf{P}(\mathcal{R}|\mathcal{H})$
×
$\mathbf{P}(\hat{\mathcal{R}}_{t+1})$
×
$\mathbf{P}(\bar{\mathcal{R}}_{t+1})$
Next robot action

Fig. 1. Flowchart representation of the synchronised grammars framework. $\mathbf{P}(\mathcal{H}_t)$ and $\mathbf{P}(\mathcal{R}_t)$ are the input probability distributions for the human and the robot predictors respectively, similarly $\mathbf{P}(\mathcal{H}_{t+1})$ and $\mathbf{P}(\mathcal{R}_{t+1})$ are the expected terminal probability distributions for the human and robot. $\mathbf{P}(\mathcal{R}|\mathcal{H})$ is the conditional probability between the terminals of the robot and the human. $\mathbf{P}(\hat{\mathcal{R}}_{t+1})$ is the influencing robot probability distribution. Finally, $\mathbf{P}(\bar{\mathcal{R}}_{t+1})$ is the distribution from which the robot draws its next action.

We further define $\mathcal{CS}_{i,s}$ as a subset of the candidate set where the next symbol to be read is terminal s, in other words:

$$\mathcal{CS}_{i,s} = \bigcup (i: \,_kX \to \lambda.s\nu \,[\alpha, \gamma])$$

At this point, it is worth reiterating that α represents the probability of the grammar generating the sequence up to the dot. Consequently, adding all the α from all states which accept s as their next terminal yields the expectation the parser has of terminal s being the next input. That is:

$$P(s_{i+1}) = \frac{\sum\limits_{i:\,_kX\to\lambda.s\nu\,[\alpha,\gamma]\in\mathcal{CS}_{i,s}} \alpha}{\sum\limits_{i:\,_kX\to\lambda.x\nu\,[\alpha,\gamma]\in\mathcal{CS}_{i}} \alpha}$$

gives us the expected probability of encountering terminal s at the next *scan* step, $P(s_{i+1})$. If we compute this probability for all terminals, we find the expected probability distribution across all terminals of the grammar. We denote the terminal probability distribution as $\mathbf{P}(\mathcal{T}_i)$.

C. Synchronisation and adaptivity

We will now present our framework to synchronise two independent SCFG parsers. Specifically we are interested in using the predictions of one parser to influence a second one. A summarised version of our algorithm is shown in Fig. 1.

Following the nomenclature established in the two previous sections, we denote $\mathcal{H}$ and $\mathcal{R}$ as the set of terminals of the first and second grammars respectively (in our experiments $\mathcal{H}$ represents the terminals of the human performer, and $\mathcal{R}$ those of the robot performer; despite this, the analysis in this section can be applied to any two SCFG parsers). Similarly, $\mathbf{P}(\mathcal{H}_t)$ and $\mathbf{P}(\mathcal{R}_t)$ represent the terminal probability distributions at step t for the first and second grammars.

By feeding the terminals to the parser and performing a full parsing step (i.e. executing *scan*, *complete* and *predict*) as well as applying the prediction method outlined in the previous section we can obtain the expected terminal probability distributions for each parser: $\mathbf{P}(\mathcal{H}_{t+1})$ and $\mathbf{P}(\mathcal{R}_{t+1})$.

We are looking for a way of influencing the second grammar. Therefore, we need to transform the first expected terminal probability distribution into a probability distribution over the terminals of the second grammar, which we will call the influence probability distribution, $\mathbf{P}(\hat{\mathcal{R}}_t)$. To achieve this we use a matrix whose elements denote the conditional probability between terminals in the first and second grammars. We designate this matrix as the synchronisation matrix, $\mathbf{S}(\mathcal{H}, \mathcal{R})$:

$$\mathbf{S}(\mathcal{H},\mathcal{R}) = \begin{bmatrix} P(r_1|h_1) & P(r_1|h_2) & \cdots & P(r_1|h_n) \\ P(r_2|h_1) & P(r_2|h_2) & \cdots & P(r_2|h_n) \\ \vdots & \vdots & \ddots & \vdots \\ P(r_m|h_1) & P(r_m|h_2) & \cdots & P(r_m|h_n) \end{bmatrix}$$
$$\text{with } \mathcal{H} = \{h_{1\ldots n}\} \text{ and } \mathcal{R} = \{r_{1\ldots m}\}$$

With the synchronisation matrix we can convert the predicted terminal probability distribution of the first grammar into the influence probability distribution using the law of total probability. This is equivalent to taking the dot product of the synchronisation matrix and the expected terminal probability distribution:

$$\mathbf{P}(\hat{\mathcal{R}}_{t+1}) = \text{normalise}\left(\mathbf{S}(\mathcal{H},\mathcal{R}) \cdot \mathbf{P}(\mathcal{H}_{t+1})\right)$$

The final step is to obtain the final probability distribution for the second grammar from both the influencing grammar and the terminal probability distribution for the second grammar. Though any method to combine two distributions into a mixture could work here, we have chosen element-wise multiplication and normalisation since it does not require any extra parameters and favours elements which have high probabilities in both distributions and punishes elements with a low probability in either distribution:

$$\mathbf{P}(\bar{\mathcal{R}}_{t+1}) = \text{normalise}\left(\mathbf{P}(\mathcal{R}_{t+1}) \odot \mathbf{P}(\hat{\mathcal{R}}_{t+1})\right)$$

Multiplying two probability distributions as above implies both of the factors have the same relative importance. Accordingly one has to take care to define the second grammar in a way allows for it to be influenced. Note that if a grammar determines its terminals with very high confidence at all times, it will not be influenced very much by other parsers.

The synchronisation matrix is the structure we use to introduce adaptivity in the framework. By modifying the elements of this matrix we can change the resulting influence probability distribution according to the preferences of each user.

Though there are several ways of altering the synchronisation matrix, we chose a punishing method whereby the user can indicate it does not like the currently selected terminals r_i, h_j. In such instances, we divide $[\mathbf{S}(\mathcal{H},\mathcal{R})]_{i,j}$ by

TABLE I
INFLUENCING (USER) GRAMMAR DEFINITION: $\mathcal{G}_\mathcal{H}$

$\mathcal{N}_\mathcal{G}$ [$\mathcal{S}_\mathcal{G}$]	{A,B,C,D,X,S,K} [S]
$\mathcal{T}_\mathcal{G} \equiv \mathcal{H}$	{a,b,c,d}
$\mathcal{N}_\mathcal{G}$ [$\mathcal{P}_\mathcal{G}$]	{S→ABXBA [1.00], X→CD [0.50], X→CXD [0.50], A→a [0.45], A→AA [0.45], A→K [0.10], B→b [0.45], B→BB [0.45], B→K [0.10], C→c [0.45], C→CC [0.45], C→K [0.10], D→d [0.45], D→DD [0.45], D→K [0.10], K→a [0.20], K→b [0.20], K →c [0.20], K→d[0.20], K→KK [0.20] }

TABLE II
BASE (ROBOT) GRAMMAR DEFINITION: $\mathcal{G}_\mathcal{R}$

$\mathcal{N}_\mathcal{R}$ [$\mathcal{S}_\mathcal{R}$]	{A,B,C,D,S} [S]
$\mathcal{T}_\mathcal{R} \equiv \mathcal{R}$	{a,b,c,d}
$\mathcal{N}_\mathcal{R}$ [$\mathcal{P}_\mathcal{R}$]	{S→S [0.2], S→AS [0.2], S→BS [0.2], S→CS [0.2], S→DS [0.2], A→a [0.8], A→AA [0.2], B→b [0.8], B→BB [0.2], C→c [0.8], C→CC [0.2], D→d [0.8], D→DD [0.2] }

a constant factor (heuristically set to *2.0* in our experiments) and renormalise the i^{th} row to add up to 1 again. This way we effectively increase the probability of all other terminals in $\mathcal{R}$ with respect to h_j.

Note that our algorithm outputs a probability distribution over the terminals of the second grammar: $\mathbf{P}(\bar{\mathcal{R}}_{t+1})$. Though taking the terminal with the highest probability as the next input would work, we choose to draw a random terminal according to the $\mathbf{P}(\bar{\mathcal{R}}_{t+1})$ distribution.

IV. SYNTHETIC ANALYSIS OF FRAMEWORK

In this section, we synthetically analyse the synchronisation and adaptivity of our framework. Before that however, we will verify that the parsing probability of a randomly generated grammars does not change significantly.

For these experiments we use two different grammars: $\mathcal{G}_\mathcal{H}$ and $\mathcal{G}_\mathcal{R}$. See tables I and II for their respective definition. $\mathcal{G}_\mathcal{H}$ encodes a sequence of terminals: {a, b, c^n, d^n, b, a} and it accepts repetition of a given terminal any number of times (with rules like A→a and A→AA). To add robustness, $\mathcal{G}_\mathcal{H}$ can also *skip* any terminal through the use of rules such as A→K, though this is left as a low-probability option. Meanwhile, $\mathcal{G}_\mathcal{R}$ is set to chose a random terminal with equal probability.

First, we wanted to verify that there are no statistically significant differences between two sets of sequences randomly generated from the same grammar. To test this we generated two sets of 1000 independent sequences each with 60 characters spawned from $\mathcal{G}_\mathcal{H}$ and obtained their Viterbi parsing probability against $\mathcal{G}_\mathcal{H}$. Spawning characters is achieved by iteratively performing a parsing step (*scan*, *complete* and *predict*); obtaining the expected probability distribution, $\mathbf{P}(\mathcal{T}_i)$; and drawing a random character from $\mathbf{P}(\mathcal{T}_i)$. Note that by Viterbi parsing probability, we refer to the *scaled Viterbi probability* which, following [11], is defined as: $v' = v^{1/l}$ where v' is the scaled Viterbi probability, v is the raw Viterbi probability and l is the length of the sequence.

We then performed a two-tailed Mann-Whitney U test on both sets of Viterbi probabilities and, as expected, found the differences *not* to be statistically significant. The median Viterbi parsing probability for the first set of sequences was 0.1395 and their inter-quartile range (IQR) 0.0292, whereas for the second set we found a median of 0.1414 and IQR of 0.0261.

A. Synchronisation test

Subsequently, we verified whether one parser can influence the state of another parser. To do so, we generated two sets of sequences from $\mathcal{G}_\mathcal{R}$; one of them by drawing the input terminal from the expected probability distribution as before. The other sequence was obtained by influencing $\mathcal{G}_\mathcal{R}$ with $\mathcal{G}_\mathcal{H}$ using our framework. Note that this effectively requires spawning an independent sequence from $\mathcal{G}_\mathcal{H}$ which again we do by drawing randomly from the expected probability distribution. The synchronisation matrix chosen here is the identity matrix (or which is the same, $\mathcal{H} \equiv \mathcal{R}$). We then parse these sequences (generated from $\mathcal{G}_\mathcal{R}$) against $\mathcal{G}_\mathcal{H}$ and obtain their scaled Viterbi probabilities, thus measuring how much did $\mathcal{G}_\mathcal{H}$ influence each sequence. We expect that the higher the influence, the higher the resulting Viterbi parsing probability and check for differences using a two-tailed Mann-Whitney U test once more.

We ran this test with two sets of 1000 sequences, each sequence being 60 characters long and found the differences in Viterbi probability to be statistically significant ($p \ll 0.01$). The median Viterbi probability of the uninfluenced sequences was 0.0762 and the IQR 0.0114 whereas for the influenced sequences the median probability was 0.1538 and the IQR 0.0315.

This confirms our claim that, at least in synthetic environments, our framework allows the state of one parser to influence the state of another parser.

B. Adaptivity test

Finally, we tested whether the synchronisation matrix can be adapted to the user's preferences. To do so, we randomly create *preferred mappings* from the $\mathcal{H}$ terminals to the $\mathcal{R}$ terminals (eg. $\{a \rightarrow b, b \rightarrow d, c \rightarrow c, d \rightarrow a\}$). The *preferred mapping* is meant to represent the different combinations of $\mathcal{H}$ and $\mathcal{R}$ a user would prefer.

Two 60 characters long sequences are generated from $\mathcal{G}_\mathcal{R}$ influenced by $\mathcal{G}_\mathcal{H}$ following our framework. The first sequence is generated by updating the synchronisation matrix (this is done by dividing the corresponding entry in the synchronisation matrix by 2 and renormalising the row, following section III-C) when the terminals do not correspond to the *preferred mapping*. For the second sequence—the control sequence—the synchronisation matrix is not updated. Note that, in both cases the synchronisation matrix is 4x4 and

Preprint version; final version available at http://ieeexplore.ieee.org
RO-MAN (2015), pp: 175-721
Published by: IEEE

initialised with all its entries to 0.25, effectively representing a random mapping between $\mathcal{H}$ and $\mathcal{R}$.

As the sentences are generated, we record the number of mismatches, that is the number of times the terminals did not correspond to the *preferred mapping*.

Repeating this process a 1000 times and performing a two-tailed Mann-Whitney U test reveals that sequences generated with synchronisation matrix adaptivity had fewer mismatches from the *preferred mappings* (median: 33, IQR: 6.0) compared to the control (median: 45, IQR: 5.25) and these results were statistically significant ($p \ll 0.01$).

This proves that, in a synthetic set-up, updating the synchronisation matrix leads to significantly fewer mismatches between the preference of the user and what the robot chooses to do.

V. Musical Human Robot Collaboration Study

The synchronised grammars framework can be readily applied to human-robot collaboration to generate music. In the following section we describe the pilot implementation of such a system.

A. Set-up

Figure 2 shows a picture the main components of our system: Reactable and Baxter.

At the heart of the system we have a Reactable, a tangible music table composed of an infra-red camera, a projector and a semi-transparent surface. Reactable's camera is able to detect fiducials placed on top of the table using the reacTIvision framework [21]. With adequate calibration, it is possible to compute the position of these fiducials with respect to the projection of the window drawn by Reactable's computer. We use a custom built application which uses the information provided by reacTIvision to drive two virtual chequerboards. Each of these chequerboards represents a musical instrument, either melody or drums. For every chequerboard there is an associated fiducial, and depending on the fiducial's position with respect to its owning chequerboard one track of music or another is played. The system is currently programmed with 4 drum patterns and 4 melody patterns. All patterns are 4 seconds long and do not change as the trials progress. Note these patterns constitute the musical primitives (or terminals) of our grammars.

Baxter is a 1.90 metres tall robot with two 7 degree-of-freedom arms, grippers on each hand and a programmable display. Baxter also receives the position of the fiducial with respect to its keyboard. By showing Baxter the kinematic configuration of the positions where each of the tracks is active, we can direct Baxter to play a specific pattern with quick movements[2]. Additionally, Baxter displays the status of the current music session on its display.

The system needs three computers, one for the Reactable, another on-board Baxter and a final one running our framework. Communication between Baxter and the main computer is done through ROS (the Robot Operating System [22]) whereas the Reactable uses a bespoke library built with Python, Unix sockets and JSON.

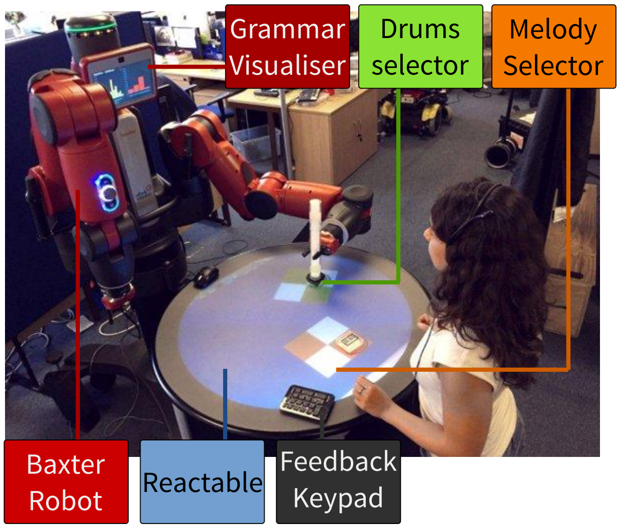

Fig. 2. Picture of our musical set-up with main components of the system.

The role of the Stochastic Context-Free Grammars is to represent the structure of the music about to be played. For this study we defined the grammars by hand (Tables I and II show the user and robot grammars respectively), but it is also possible to generate these grammars from expert demonstrations using the method presented in [11].

The initial synchronisation matrix was obtained by asking a participant to generate music with our system during 5 minutes and then counting the number of co-occurrences for each user and robot terminal. The participant controlled both the melody and the drums and thus there was no robot involvement. The resulting synchronisation matrix was the starting matrix for all participants and is shown below:

$$\begin{bmatrix} 0.37 & 0.05 & 0.27 & 0.31 \\ 0.23 & 0.04 & 0.46 & 0.27 \\ 0.03 & 0.08 & 0.83 & 0.06 \\ 0.08 & 0.82 & 0.06 & 0.04 \end{bmatrix}$$

Eight participants, two of them female, aged 21–34 took part in our pilot study. Each participant had 4 trials to create 3 minutes of music with Baxter. The first of these trials was discarded as training. Participants could also indicate the system the track Baxter had selected was not an appropriate match for their own choice of track. This assessment was made by pressing the Enter key on the feedback keypad situated next to them. Note the changes made to the synchronisation matrix were kept across trials (except for the training trial).

At the end of the trials, participants were asked to complete a questionnaire with the following questions in a 5-point Likert scale:

- Performing music with Baxter was difficult (*difficult*).

[2] A previous version of this system used inverse kinematics, but that was found not to be responsive enough.

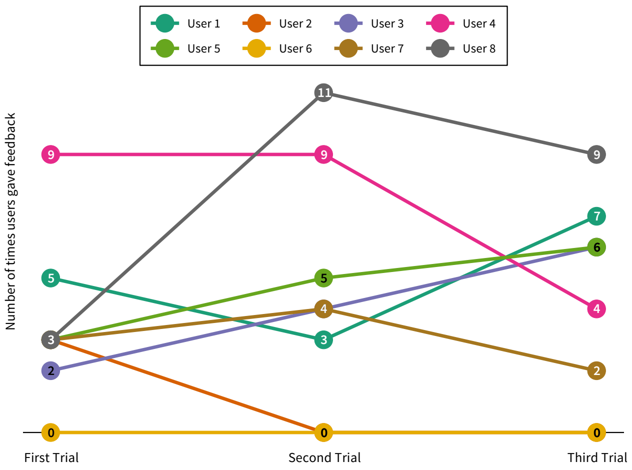

Fig. 3. Number of times users gave feedback to the system by participant.

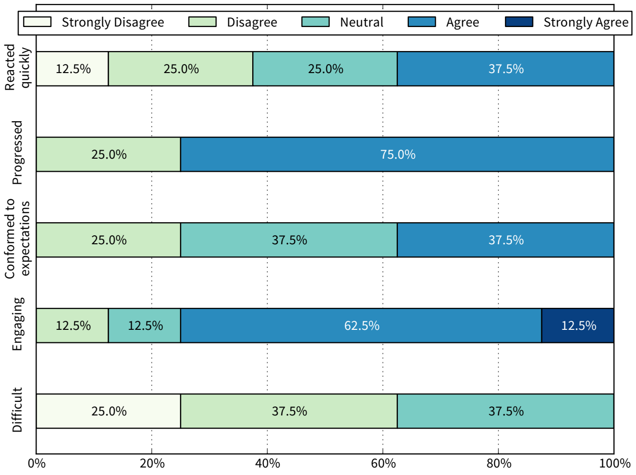

Fig. 4. Results of the questionnaire filled in by participants after the trials.

- Performing music with Baxter was engaging (*engaging*).
- Baxter's actions conformed to my expectations (*conformed to expectations*).
- Baxter became a better accompanist as trials progressed (*progressed*).
- Baxter reacted quickly to my changes in the melody (*reacted quickly*).

B. Results

Figure 3 shows the amount of negative feedback per participant across the three trials. It can be seen in the figure that 4 participants (S2, S4, S6 and S7) decreased the overall amount of negative feedback. In contrast, participants S1, S3, S5 and S8 increased the overall amount of negative feedback. S6 did not provide feedback at all.

The final synchronisation matrices from two participants are shown below. We remark that both matrices are different from each other as well as from the original synchronisation matrix.

$$\underset{a)}{\begin{bmatrix} 0.53 & 0.06 & 0.19 & 0.22 \\ 0.37 & 0.01 & 0.18 & 0.44 \\ 0.01 & 0.08 & 0.80 & 0.11 \\ 0.05 & 0.54 & 0.32 & 0.09 \end{bmatrix}} \qquad \underset{b)}{\begin{bmatrix} 0.18 & 0.01 & 0.52 & 0.29 \\ 0.24 & 0.00 & 0.47 & 0.28 \\ 0.01 & 0.08 & 0.89 & 0.02 \\ 0.19 & 0.47 & 0.03 & 0.31 \end{bmatrix}}$$

Figure 4 shows the results of the questionnaires. 3 people agreed the robot *reacted quickly* while 3 people disagreed or strongly disagreed with the statement. 6 people agreed the robot improved as trials *progressed*. 2 people disagreed that the robot *conformed to their expectations* whereas 3 agreed that was the case. 6 people agreed or strongly agreed the task was engaging. 3 people were neutral about whether they had found the task *difficult*, the rest disagreed or strongly disagreed.

C. Discussion

We posit the differences in the amount of negative feedback stem from the fact that every participant has a different expectation of the system's learning curve. This is substantiated by the questionnaires where the answers to whether the robot *conformed to expectations* were evenly spread. It is possible as well that different users require different updating constants in the synchronisation matrix.

The sample final synchronisation matrices confirm that our framework can adapt to the preferences of different users. For instance, the first robot terminal in $a)$ gives most of the probability mass to the first human terminal, whereas $b)$ gives it to the third human terminal. Moreover, observe how these preferences are not present in the original synchronisation matrix.

With respect to the questionnaires, we were not surprised to find that most users agreed the task was engaging since, as we mentioned in the introduction, most people enjoy music. More interesting was the fact that most people (6 out of 8 participants) felt the robot had become a better accompanist as the trials progressed. We remark that this is a similar ratio to the number of people which decreased their overall negative feedback (4 out of 8). This similarity encourages us to carry out further experimentation to establish statistical significance.

There is no clear consensus with respect to whether the robot reacted quickly to user's actions and our data shows that there were network-induced delays in Baxter's actions. This may also explain the divergent answers to whether the robot conformed to the user's expectations.

VI. Conclusion and Future Work

We have presented a framework to synchronise two Stochastic Context-Free Grammars. Our framework allows to account for the expected values of another parser. This can be useful in many scenarios as it provides a formal method to combine the constraints given by the structure of one task and the needs of another independent—but simultaneous—task. With our framework it is possible to incorporate external feedback or ignore it due to task constraints. The synchronised grammars framework further allows for personalisation. This is achieved through the synchronisation matrix. The synchronisation matrix is the link between the two parsers and changing it can give rise to a wide array of behaviours.

Our synthetic experiments confirm both the ability to let one parser influence the other as well as the adaptivity properties of the synchronisation matrix.

All the algorithms we used are probabilistic, which lets the framework provide a natural approach—by randomly drawing from a probability distribution—for a robot to play music with variations which reflect the user's preference. This is relevant to a number of domains. Amongst them is music, where small variations are vital for the music not to sound artificial.

Music usually has an internal structure and users have very varied tastes. With its ability to synchronise to external patterns and adapt to users, our framework fits well with musical human-robot collaboration. To test this, we built a system with a tangible music table and a two-armed robot. From the results it was observed that the robot could be influenced by the music the users were generating as well as to adapt to feedback by users.

Specifically, the results of our pilot study suggest that, given enough time, the synchronisation matrix will converge to the preferences of the user. This was corroborated by both the user's questionnaires where 6 out of 8 participants agreed the robot had improved across the trials and the results from the synthetic experiments where the mismatch rate was significantly lower with an adaptive synchronisation matrix.

The implementation of our system can be improved through reworking the network connectivity between Reactable, Baxter and the computer running the synchronised grammars framework, which leads to latency in the robot's actions. This issue was highlighted by the mixed results users gave when asked about Baxter's reaction speed.

In the future, we plan to perform in-depth experiments to corroborate our pilot study results in more complex scenarios. We further aim to investigate how to extend the synchronised grammars framework to work with more than two grammars, thus bringing the benefits we describe in this paper to multi-robot collaboration. Another interesting avenue is to apply the synchronised grammars framework to other tasks such as collaborative assembly.

Acknowledgment

The authors would like to thank all the members of the Personal Robotics Lab, particularly Arturo Ribes for writing the Reactable application, Ayşe Küçükyılmaz for conducting the pilot study and Jonas Golland for composing the music samples.

This work was supported in part by the EU FP7 project WYSIWYD under Grant 612139.

References

[1] M. Cicconet, M. Bretan, and G. Weinberg, "Human-Robot Percussion Ensemble: Anticipation on the Basis of Visual Cues," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 105–110, Dec. 2013.

[2] C. Crick, M. Munz, and B. Scassellati, "Synchronization in Social Tasks: Robotic Drumming," in *Proceedings of the International Symposium on Robot and Human Interactive Communication*. IEEE, Sep. 2006, pp. 97–102.

[3] K. M. Kitani and H. Koike, "ImprovGenerator : Online Grammatical Induction for On-the-Fly Improvisation Accompaniment," in *Proceedings of the Conference on New Interfaces for Musical Expression*, no. 1, Jun. 2010, pp. 469–472.

[4] D. Quick and P. Hudak, "A Temporal Generative Graph Grammar for Harmonic and Metrical Structure," in *Proceedings of the International Computer Music Conference*. ICMA, Aug. 2013, pp. 177–184.

[5] M. P. Michalowski, S. Sabanovic, and H. Kozima, "A dancing robot for rhythmic social interaction," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, Mar. 2007, pp. 89–96.

[6] G. Hoffman and G. Weinberg, "Gesture-based human-robot jazz improvisation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2010, pp. 582–587.

[7] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," in *Proceedings of the Meeting of the Association for Computational Linguistics*. ACM, Jul. 2003, pp. 423–430.

[8] Y. Sakakibara, M. Brown, R. Hughey, I. S. Mian, K. Sjölander, R. C. Underwood, and D. Haussler, "Stochastic context-free grammars for tRNA modeling," *Nucleic Acids Research*, vol. 22, no. 23, pp. 5112–5120, Nov. 1994.

[9] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password Cracking Using Probabilistic Context-Free Grammars," in *Proceedings of the IEEE Symposium on Security and Privacy*, May 2009, pp. 391–405.

[10] Y. A. Ivanov and A. Bobick, "Recognition of Visual Activities and Interactions by Stochastic Parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, Aug. 2000.

[11] K. Lee, Y. Su, T.-K. Kim, and Y. Demiris, "A syntactic approach to robot imitation learning using probabilistic activity grammars," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1323–1334, Dec. 2013.

[12] T. Fong, C. Thorpe, and C. Baur, "Collaboration, dialogue, human-robot interaction," in *Robotics Research*, ser. Tracts in Advanced Robotics. Springer, Oct. 2003, vol. 6, pp. 255–266.

[13] G. Hoffman and C. Breazeal, "Cost-based anticipatory action selection for human-robot fluency," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 952–961, Oct 2007.

[14] J. Shah, J. Wiken, B. Williams, and C. Breazeal, "Improved human-robot team performance using chaski, a human-inspired plan execution system," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, Mar. 2011, pp. 29–36.

[15] K. Ebcioğlu, "An Expert System for Chorale Harmonization," in *Proceedings of the AAAI*, Aug. 1986, pp. 784–788.

[16] A. Tidemann and Y. Demiris, "Groovy Neural Networks," in *Proceedings of the European Conference on Artificial Intelligence*, vol. 178. IOS Press, Jun. 2008, pp. 271–275.

[17] Z. Zhang, T. Tan, and K. Huang, "An extended grammar system for learning and recognizing complex visual events." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 240–255, Feb. 2011.

[18] N. Chomsky, "Three models for the description of language," *IEEE Transactions on Information Theory*, vol. 2, no. 3, pp. 113–124, Sep. 1956.

[19] A. Stolcke, "An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities," *Computational Linguistics*, vol. 21, no. 2, pp. 165–201, Jun. 1995.

[20] J. Earley, "An efficient context-free parsing algorithm," *Communications of the ACM*, vol. 13, no. 2, pp. 94–102, Feb. 1970.

[21] M. Kaltenbrunner and R. Bencina, "reacTIVision: A Computer-Vision Framework for Table-Based Tangible Interaction," in *Proceedings of the International Conference on Tangible and Embedded Interaction*. ACM, Feb. 2007, pp. 69–74.

[22] M. Quigley, B. P. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS : an open-source Robot Operating System," in *Proceedings of the Open Source Software Workshop at the International Conference of Robotics and Automation*, May 2009.