LL: LISTENING AND LEARNING IN AN INTERACTIVE IMPROVISATION SYSTEM

Nick Collins
University of Sussex
N.Collins@sussex.ac.uk

ABSTRACT

Machine listening and machine learning are critical aspects in seeking a heightened musical agency for new interactive music systems. This paper details LL (ListeningLearning), a project which explored a number of novel techniques in this vein. Feature adaptation using histogram equalisation from computer vision provided an alternative normalization scheme. Local performance states were classified by running multiple k-means clusterers in parallel based on statistical summary feature vectors over windows of feature frames. Two simultaneous beat tracking processes detected larger scale periodicity commensurate with bars, and local IOI information, reconciling these. Further, a measure of 'free' playing as against metrically precise playing was explored. These various processes mapped through to control a number of live synthesis and processing elements, in a case study combining a human percussionist and machine improvisation system. A further project has subsequently adapted core parts of the work as a Max/MSP external, first used for Sam Hayden's violectra project, and now released in conjunction with disclosure of code for this paper.

1. INTRODUCTION

We hope to build artificially intelligent musicians which can engage as active and equal participants in musical interaction [1, 2]. Virtual musical agents can interface to the real world through audio input and output, requiring a realtime artificial ear to interpret audio input, live sound synthesis to create output on the fly, and musically respectful decision making inbetween. Such systems can be placed in art installations, in gaming, or in other activities including therapeutic and pedagogical settings. The primary motivation and ultimate test of this research project concerns the challenge of concert conditions.

There is a long and fascinating history to machine interaction in performance, from such 1960s and 1970s precedents as the analogue machine listening pieces of Sonic Arts Union composers Gordon Mumma and David Behrman [3] to the timbre-driven improvisation systems currently being developed by Bill Hsu [4] or the computerized online structure formation of OMax [5]. The latter is one of a number of recent systems exploring the use of machine learning techniques for virtual musicians (see [6] for a review).

The current project is referred to as ListeningLearning (LL), though apt anagrams include 'I'll sing, inner agent', 'sing, eternal lining', and 'tell insane ringing'. LL was constructed for a free improvisation setting, with a primary emphasis on timbral and rhythmic alignment between virtual and human musician. Whilst many aspects of the system are generally adaptable to multiple contexts, the specific concert study for which the system was originally built was a collaboration with experienced percussionist improviser Eddie Prévost, organized through the auspices of the Live Algorithms for Music network in the UK. This original LL is for human on drum kit, and musical AI.

In engineering the system, the design had to confront many cutting edge issues in realtime control and signal processing. The uses of machine learning extends from signal level and frame-based features to operations on derived symbolic attributes. Although some aspects of the machine listening and learning algorithms are rather technical, the exposition below tries to dwell on critical points to enable expert understanding, without becoming bogged down in the mathematics and particular signal processing code. The author is happy to share the source code with interested parties where full disclosure is necessary to full understanding, in line with the paper plus source model of academic inquiry. A Max/MSP external (ll~) has also been created for further project tests and a general release to accompany this paper.

We proceed by revealing technical and compositional aspects of the system itself, before a discussion of evaluation in rehearsal and concert. We end by discussing future work in interactive systems building.

2. SYSTEM OVERVIEW

Subsisting entirely on the signal collected from a microphone, LL must be able to cope with fluid and mutually influential improvisation with a human partner. Figures 1 and 2 give an overall picture of the modules within the system, and the sites of specific technologies to be further elucidated in the upcoming subsections. Figure 2 focuses in on the machine listening apparatus, including some aspects not covered in the broad overall input/output system architecture of Figure 1. Many modules were developed as generally reusable for future systems building, but the hard deadline of a concert, and knowledge of the specific setting of human drum kit player plus machine, focused

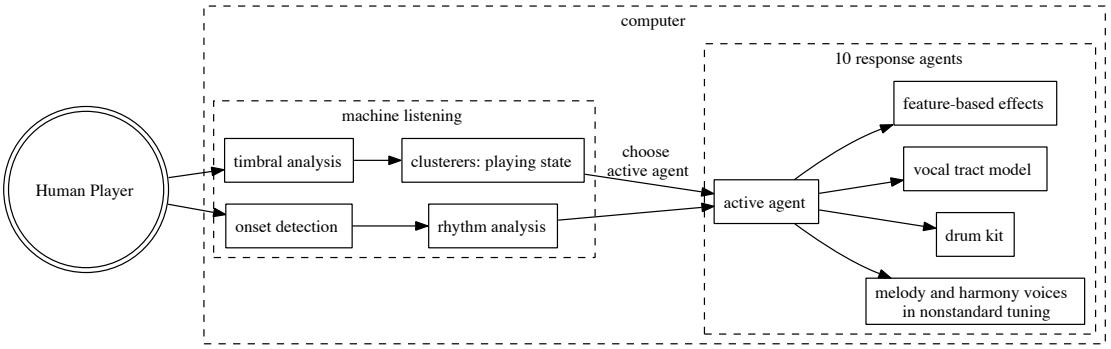

Figure 1. ListeningLearning (LL) system overview.

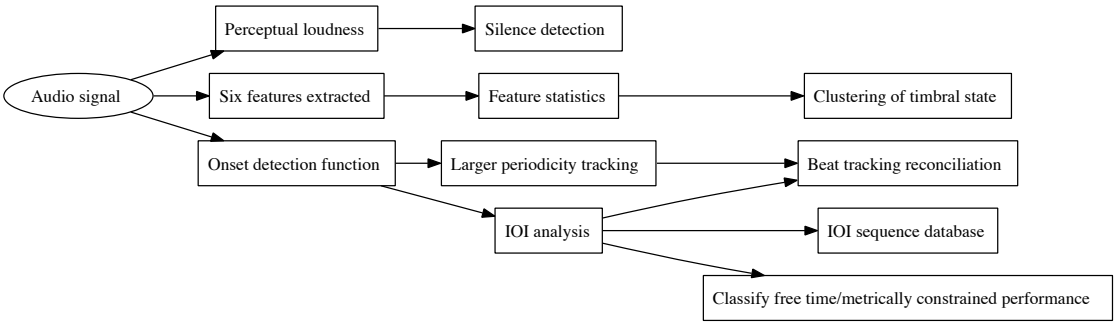

Figure 2. A closer look at the machine listening modules.

development work in the run up to the premiere.

Where machine learning facilities are incorporated, whether adapting online or offline, system state can always be saved and loaded, facilitating rehearsal training ahead of a concert, and potentially concert to concert development.

2.1 Features

The feature extraction paradigm underlies many strands of current computer music research, though the most fervent technical developments in recent years are perhaps traceable to the commercial crossover of music information retrieval. In interactive systems, features appear from concatenative synthesizers [7] to adaptive effects [8] and feature-tracking synthesis [9, 10].

Machine learning algorithms are more tractable when they operate on a smaller set of inputs (corresponding to a lower dimensional space). Features provide a form of data reduction, focusing in on essential attributes. Whilst any such reduction may lose information, features are selected to provide insightful control signals summarizing important aspects of the source signal. In the extreme, a whole piece might be summarized by a single value (e.g., a categorization as a Schubert impromptu for piano, or the identification of a single global tempo of 156bpm), and though there are consequences to dropping time series information in exchange for global statistics, a hierarchical nesting of features at multiple scales may be highly valuable.

In the system described here, feature vectors are formed for successive windows of samples, with a basic hop size of 512 and window size of 1024, the system operating at the standard 44100 sampling rate. This works out to approximately 86 feature frames per second. The features are formed in the time, complex exponential (DFT spectrum), and cosine function (Discrete Cosine Transform) spaces, obtaining single values for the given window. Because of the free improvisation percussion setting, timbral and rhythmic features are prioritized in the system, and no pitch information treated, reminiscent of a similar decision in work by Bill Hsu [11]. The six basic features are listed in Table 1.

In order to use the features in a comparable way within a feature vector, some form of normalization is required. Tactics would include a simple min-max normalization, a Gaussian assuming standardization (keeping signal values within say three standard deviations of the mean), and hard-coded transforms particular to the type of feature (for instance, a mapping from frequency to a logarithmic pitch space, then normalization). For this project, each feature is run through an adaptive distribution model, which tracks the range and distribution of feature values, providing a

#	Origin	Feature
1	Discrete Cosine Transform (DCT)	80% percentile of cosine basis energy
2	DCT	Inter-frame flux, cosine-wise
3	Time domain	RMS amplitude
4	Discrete Fourier Transform (DFT)	Spectral centroid
5	DFT	Spectral irregularity (inter-bin power fluctuations)
6	DFT	95% percentile of spectral energy

Table 1. The six basic features extracted.

more complicated adaptive normalization process. A similar technique has been explored in computer vision algorithms as 'histogram equalisation' [12, p. 188]. In LL, the feature distributions can be obtained in training and then fixed, or developed online, though there is a transient behaviour in the construction of the mapping particularly in initial start up. [1] Figure 3 shows the effects of the histogram equalization versus a max-min normalization; the features in the former case are compressed in a way that focuses on detail wherever particular values are more frequently encountered.

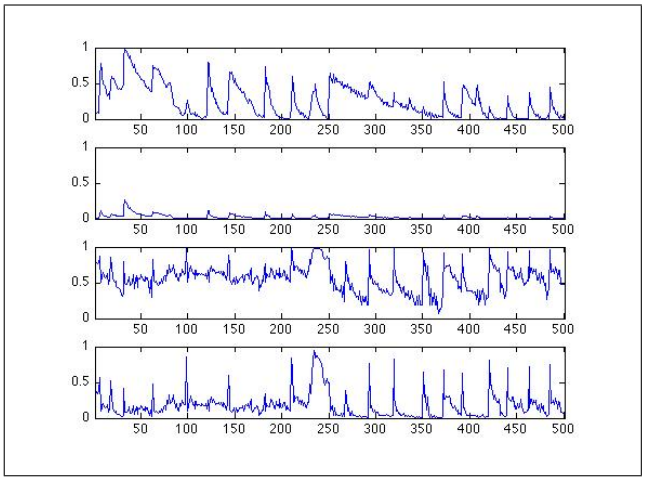

Figure 3. Feature trails and adaptation results. The top two lines show the results for histogram equalization (first line) and normalization (second, overall max occurring outside plotted window), for RMS amplitude. The bottom two show another comparison, this time for the 95% energy point in the power spectrum. The features arose from analysis of a drum kit audio signal from an overhead microphone. It can be observed that the adaptive feature normalization makes better use of all the available [0,1] range, supplying a maximization of dynamic range. Although there is a possibility of increasing noise, the assumption is that all features utilized provide healthy and interesting viewpoints on the musical signal, and this normalization technique maximizes the value of the information provided.

2.2 Feature vector sequence modeling and clustering

We would like to have an unsupervised training procedure to adapt to the timbral choices of a given performer, and enable the computer to respond differentially to different modes of performance. A precedent here is work by Dannenberg, Thom and Watson [13], who trained the computer to distinguish eight distinct playing styles from 13 MIDI features through clustering. Clustering algorithms provide the unsupervised learning facility here too, with the decision based on the timbral content within the last one and a half seconds. In order to make use of the feature data over longer timescales, but keep model complexity (dimensionality) down, the feature vectors from window to window are too plentiful. This reduction is effected by modeling for a subset of features, the feature time series as statistics. In this case, the mean and range are taken, though there are many other modeling options (such as envelope (curve) fitting to segments, tagging by index into a shapes database). [2]

The reduction here supplies the input for clusterers, which are used to differentiate different timbral playing states. Each clustering model is built using longer term statistics over a subset of two features of the original extraction process. Every ten feature frames, for each of the two features, a mean and a range are taken. This reduces from 20 to 4 values. A new summary feature vector corresponding to the last half second or so is formed by looking at a memory of the last five reductions, giving 20 values in the place of 100. These twenty values are the input to a clustering model, representing feature activity within the last 50 feature frames, or around 580 milliseconds.

Many clustering models were tried, including both online and offline agglomerative clustering [15, 16]. It was eventually found that the most robust method was not to train clustering online; an online algorithm leads to inconsistent changing indexing decisions over time, despite the ostensible notion of adapting to performance. The clusterer was instead trained at one point in time having gathered data. For instance, five minutes of rehearsal playing by a musician would supply 25839 feature vectors, and thus by the reduction above 2583 twenty-dimensional points. An effective method was to train multiple standard k-means clusterers over ten iterations, with the Euclidean metric for distance. Each clusterer was initialized to random starting cluster positions within the permissible range, and the 'best' taken with respect to some error criteria. In practice, it was found that ten clusters provided a good target number of states for live performance, and ten 10-means clusterers were trained and the best selected based on per data point error. [3]

In order to avoid an unstable jumping of state in the output, a consistency test was effected by taking a majority decision over the last ten clusterer outputs. A clusterer was invoked around every ten frames (every 116 milliseconds), and one majority decision taken around each second. Reflecting data from the last second and a half or so, this made for a more stable decision, though still on the order of human reaction times in improvisation. Such reactions are

[1] This motivates a guard region, for example, waiting for a minute for feature adaptation to settle down, but is not a problem if rehearsal and soundcheck time is available, or even if the system can adapt in concert during someone else's prior piece!

[2] See also Joder, Essid and Richard [14] on comparing various temporal integration methods in a music information retrieval framework

[3] Following the lead of Hazan et al. [17], the Akaike Information Criterion is one test which offsets the number of clusters (size of k) against the assignment error per data point.

reasonable for an emulation of human like playing.

Two separate clusterers are run in parallel in the concert system, based on features 1 and 2, and 3 and 4 respectively. The timbral state decisions from the clustering then feed through to the musical decision centres of the artificial musician.

2.3 Rhythmic analysis: larger scale periodicity tracking, and IOI analysis methods

Whilst timbral state tracking is one aspect to tracing the local manner of performance, rhythmic information was dropped in coming to the clustering decision. Parallel mechanisms have been effected to track periodicities in the input signal. There are two rhythmical modeling components, one based on analysis of Inter Onset Intervals (IOIs), and another on larger scale (one to three second or so) repetitions commensurate with a cyclic repeat at the level of a measure.

To explain the latter first, an onset detection function is formed via a frequency warped spectrum. 40 Equivalent Rectangular Bandwidth scale bands bunch up bins from the FFT power spectrum, and are passed through an equal loudness contour correction function [18]. 40 derivative features are created, where the first four are the powers in the bands aggregated (summed) in tens (0-9, 10-19,20-29,30-39), and the remaining 36 are differences between power now within the four aggregate bands, and that from one to nine frames ago in time, bandwise.

The periodicity detector returns a decision once per second, based on running totals for each potential periodicity, each updated at its own cycle period. A variant of autocorrelation is used, with larger periods than standard beat tracking, from 116 to 230 frames corresponding to four beats at from 180 to 90 bpm respectively. An allowance for the best match within +-3 frames copes with limited expressive timing variation; matches are scored by finding the minimal featurewise difference sum. The negative of these values (favouring minimal difference) are summed over the duration of a cycle, normalized by cycle length, and compared between cycles on the basis of the largest score. Both the winner and runner up period are sent on to the next stage of analysis.

Meanwhile, an IOI analysis beat tracker is running in parallel, as per IOI histogramming for beat tracking [19], with some novel aspects now described. IOIs are collated within a window of the last three seconds, and an analysis proceeds once per second. For robustness to swing patterns and other typical rhythmic spacings in the analysis, the original IOIs are combined with merged pairs of successive IOIs. This makes sure that more potential IBIs (inter beat intervals) are present in the mix to be analyzed.

At the core of the analysis, we seek to differentiate free time playing from steadier metrically established performance. This is seen as a critical aspect of free improvisation performance, particularly in the case of working with a free ranging drummer! The algorithm for differentiation of the two states intuitively runs:

1. Find the list of IOIs in a window of recent history
2. Sort the list of occurring IOIs into order of size of interval
3. Starting at each IOI in the list in turn, seek out the length of the subsequence of successive IOIs within a tolerance of 50 milliseconds of the starting point. The length of the subsequence starting at each IOI reflects the number of close companions, allowing for expressive timing in a kinder way than imposing histogram boundaries (which may split closely related IOIs by bucket divisions). The array of subsequence lengths is scanned to determine peak areas in the distribution of IOIs.
4. Make a decision based on whether only a few peak areas appear, with simple hierarchical duple or tuple relationships, indicative of a more metrical basis, or whether a wider distribution of IOIs is present commensurate with freer playing.

If free time playing is indicated, an immediate prediction model is used for future IOIs based on data collected in free time situations of the next IOI given the last two (this Markovian prediction model is constantly being trained in any given performance). Otherwise, a beat based model is implicated, with the winning IBI (Inter Beat Interval) the preferred tempo within the IOI analysis which is corroborated as a divisor of the longer term periodicity analysis. The phase is determined by a brute force search for the phase divisor of the IBI which best matches the window of recently observed IOIs.

2.4 Response deliberation and generation

Through the benefits of the machine listening analysis, LL has available to it timbral state, and rhythmical structure data to guide responses. The original LL performance system was conceived as being a hive of ten agents, one for each timbral state, each with different takes on the available synthesis and processing resources. Four main modules supply independent response generation:

1. Up to four voice harmony played with a digital model of an analogue subtractive synthesizer, with a different tuning system for each agent
2. A physical model of the vocal tract, being a one dimensional succession of linear tube sections with scattering junctions [20], with control model establishing synthesis gestures over time
3. Sample based kits, one for each agent, with an algorithmic drumming routine based on the listening rhythm model
4. Feature-based effects [8] where the form of live processing of the musicians drum kit is influenced by feature data captured in contemporaneous machine listening. Fifty effects units were algorithmically generated, based on filters, phasing, flanging, delays and distortions, all parametrised by different live listening features, including the use of onset detector triggers.

The use of ten agents was justified in establishing a sufficient diversity of responses for the open-ended improvisation planned for the concert. The agents differed in their predilections to action for both the interpretation of the rhythmic analysis (some were more prone to 'want' to be free) and restraint of response generation (for example, more aggressively interpreting a sparsity of observed onsets as an excuse to itself pro-actively generate materials, or treating human silence as an important cue to themselves rein in activity). Scare quotes are appropriate here; whilst they may invoke the programmers intentions, anthropomorphic agent instance variables named 'slavishness', 'consistency', or 'steadfastness' (as appear in LL's source code) are not in themselves sufficient to attribute the machine with more highly developed cognitive agency in decision making, but only as a guide to programmer intentions [6]. The rule based logic within the 'brain' of each agent was sufficient to determine activity levels for each of the four modules of output in response to an observed musical environment.

2.5 Implementations

In the original system, listening and learning facilities were split across an independent C++ application, and a SuperCollider program. Whilst the longer term periodicity detection runs ERB band based features in its onset detection function, the onset events for IOI analysis are determined using an onset detection algorithm attributable to Dan Stowell [21]. The two applications inter-communicate using Open Sound Control, and the IAC bus is used to send MIDI messages from SuperCollider to Logic MainStage for the sample based drum kits. The reason for building a separate C++ application rather than building all signal processing work into SC plug-ins, was the complexity of the entangled machine learning apparatus, and the need for saving and loading of a complex system state between sessions.

Core components (the feature adaptation and clustering, though without the beat tracking modules) have also been built into a Max/MSP external, ll~. The external is released to accompany this paper; Figure 4 shows a screenshot. In this ll~ formulation, some hard-coded parameters of the original system have been opened up, and the ERB-based onset detection function is also made available as a feature. Two arguments allow first setting the size of the windows over which features are summarised, and second the number of windows over which a majority decision is taken; the external must also have a mono audio input. Messages are passed to save or load state, reset the clusterer to collect new data, or hunt for a specified number of clusters. The outputs are the current state from the clusterer, how full the memory buffers are with collected feature sequence data, and the 14 feature sequence means and ranges themselves (2 values for each of 7 features), readily mapped through to control other processes. [4]

[4] A user can set the window size to one frame to just get back the original features if a direct feature extractor is desired

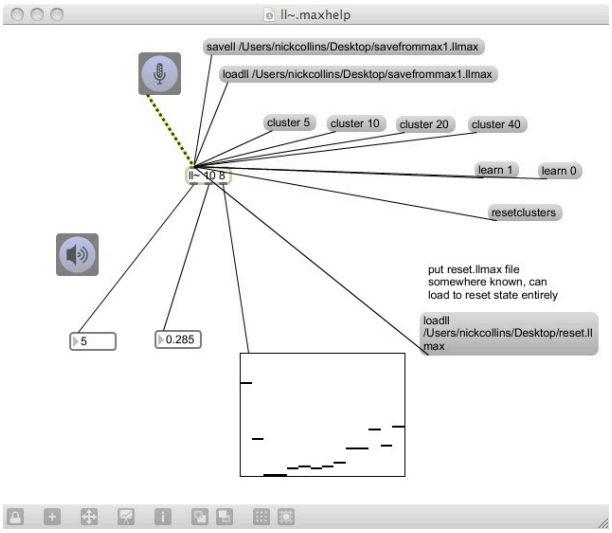

Figure 4. ll~ Max/MSP external

3. EVALUATION

The original LL system had its premiere in 2009, and the adapted technology has been tested in a further project through 2010. Since in the moment quantitative evaluation methods in HCI are only at a tentative stage [22], HCI methodologies for feedback from rehearsal or concerts tend to be based around more qualitative methods of review [23, 24], and the evaluation presented here is relatively informal, based on in rehearsal discussion, post concert analysis, and feedback from development of the external.

LL as a self-contained performance system had its live premiere in front of an audience of around one hundred people, at Cafe Oto in London on August 6th 2009. The percussionist Eddie Prévost took an active part in playing with a variety of systems, as part of the Live Algorithms for Music Songs for Dynamical Systems event. [5] Prior to the Thursday night concert, participants in a LAM workshop had been developing systems and rehearsing with Eddie since the Monday. Unable to attend the workshop component, I had been preparing the LL system for around three months before the concert, beginning with more general coding, and focusing in on the specific LAM event when kindly invited to participate. Anticipating limited rehearsal time, I had simulated the likely nature of inputs beforehand via canned drum recordings and live beat boxing input. Despite some pressures around the event, the system performed as expected in the concert.

I had an hour of so of rehearsal time with Eddie on the previous afternoon to make some recordings for overnight training purposes, and to run the system live for familiarization in the interactive setting. We had chance to compare using a version of the system trained on Eddies playing, with a prior saved system trained on my own vocal improvisation. This was further explored the next day in afternoon rehearsals at the venue itself, when we settled on using the system trained on my vocals as the prior for

[5] http://www.cafeoto.co.uk/SongsForDynamicalSystems_000.shtm

the in gig adaptation to Eddie. He preferred the thought of multiple contributors this implied, instead of basing all analysis on just himself.

A further interesting outcome from rehearsal concerned the question of discerning freer rhythmic playing from the invocation of steadier pulsation. Eddie had not considered this dichotomy ahead of my requests; we found that when asked to perform separately in the two ways in rehearsal, to help with testing the algorithms, elements of metricality underlined parts of his 'free' playing and vice versa. He appreciated the task as an interesting reflection on his practice, though the speed with which he shifted between establishing pulse and breaking it down again provided a challenge to the categorizing algorithm.

Two noteworthy things occurred on the concert night as reflections on the system. The first was that Eddie ended up playing with the system for around seven minutes, rather than the three minutes planned! The second was that he came offstage brimming with enthusiasm, and happy to confirm his enjoyment of the experience. However, this may be attributable both to adrenalin, and to a desire to maintain enthusiasm in the presence of the system's specific author and others involved with the concert pieces that night.

In order to approach the system in a more neutral light, a concert recording was invaluable for a more sober debriefing. Eddie was understandably cautious in his approach to a recorded document of improvisation, and our discussions tended more to general feedback on his feelings on the whole event. His approach to interacting with the various systems at the LAM event had been to treat the encounter with 'pretty much the same manner as I may deal with a musician'. He tried to avoid bias, and concentrate on the specific musical exchange regardless of background questions of the social capability, autonomy, alienness and more of any computer musical constructs. He recognized that the systems reflected the preparations of their programmers, and machine learning notwithstanding, was concerned 'how far can a dialogue between two human beings e.g., the programmer and the active musician be mediated successfully in this manner'. He reflected that his measure of a system which was not suffering a 'limit on mutuality' was that it should 'respond to things which cannot be anticipated'. And for successful improvisation, like arresting conversation, the endpoint should not be predictable: 'The outcome of a dialogue ought not (I suggest) to be something that can easily be anticipated'.

Subsequent to this performance, the feature adaptation and performance state clustering components of LL were turned into a Max/MSP external. This was undertaken for the composer Sam Hayden's AHRC funded project, 'Live Performance, the Interactive Computer and the Violectra', specifically for work on incorporating additional facility to the computer part as an autonomous improviser. Interestingly, the electric violin being tracked is a pitched instrument, unlike the previous percussion work, but the timbral features (which in part resolve some spectral resonances) still give an angle on performance states of the system. Sam provided a large amount of feedback on the development of the external, testing it repeatedly in rehearsal and performance. A publication about this project from his own perspective is under review [25]. A primary observation was his encounter with the tradeoff of trusting the machine, versus controlling every facet; the external gave a desired agency to the system, if obscuring certain more linear relationships previously in place. The external has the ability to save and load on the fly, recalling state spaces established earlier in a concert, or on a different day, even across instrument training sessions, potential yet to be fully exploited. Following this project's completion, the external is now ready for release to a wider user base for more general testing.

4. CONCLUSIONS

The ListeningLearning system described in this paper provides a snapshot of work incorporating enhanced machine listening and machine learning capabilities into interaction. Although the original LL system is but one working model of music — as Eddie Prévost found, one mediation from programmer to musician — there is great opportunity for learning machines to step somewhat outside of their initial programming, even if we may also express a skepticism that the bounds of the space within which learning takes place is itself constrained [26]. The capacity to save and load state between rehearsal and concert, and mix and match trained states, potentially allows a longer-term development for the system across multiple rehearsals and performances. This scope for continual adaptation is one of the most exciting future opportunities for development.

In this sense, LL is but an initial draft for an artificial musician which can develop through a series of rehearsals and concerts. The ultimate conception may be that of a *musical familiar* that adapts with a musician from childhood lessons into adult performance, developing itself as they grow. Eddie reacted with enthusiasm to the idea of such as system to train and play with over the longer term: 'This I like the sound of. And, hope that I get to meet.' This comment though also indicates that LL and the circumstances of its deployment had not gone far enough, and motivates future plans. Building such systems, and evaluating them through longitudinal studies, will not be easy. The attribution problem in machine learning notes the difficulty of assigning credit to guide the learning of a complex system, particularly when praise or negative feedback are themselves scarce [6].

Nonetheless, with the release of the ll~ external, more qualitative feedback can be gathered. Many of the techniques employed in this paper, from multiple beat tracking mechanisms, through feature adaptation, to mappings from clusters to performance states, may prove viable as the basis of further systems.

5. ACKNOWLEDGEMENTS

Hearty thanks to Ollie Bown, for his efforts in organizing and documenting the summer 2009 LAM workshops and concert, to Eddie Prévost for his indispensible musical contribution, and to Sam Hayden as the rightfully demanding

commisioner of ll˜.

6. REFERENCES

[1] R. Rowe, *Machine Musicianship*. Cambs, MA: MIT Press, 2001.

[2] N. Collins, "Musical robots and listening machines," in *Cambridge Companion to Electronic Music*, N. Collins and J. d'Escriván, Eds. Cambridge: Cambridge University Press, 2007, pp. 171–84.

[3] J. Chadabe, *Electric Sound: The Past and Promise of Electronic Music*. Englewood Cliffs, NJ: Prentice Hall, 1997.

[4] W. Hsu, "Two approaches for interaction management in timbre-aware improvisation systems," in *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, August 2008.

[5] G. Assayag, G. Bloch, M. Chemillier, A. Cont, and S. Dubnov, "OMax brothers: a dynamic topology of agents for improvization learning," in *AMCMM '06: Proceedings of the 1st ACM workshop on audio and music computing multimedia*, 2006, pp. 125–132.

[6] N. Collins, "Reinforcement learning for live musical agents," in *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, August 2008.

[7] M. Casey, "Soundspotting: A new kind of process?" in *The Oxford Handbook of Computer Music.*, R. Dean, Ed. New York: Oxford University Press, 2009.

[8] V. Verfaille and D. Arfib, "A-DAFx: Adaptive digital audio effects," in *International Conference on Digital Audio Effects (DAFx)*, Limerick, 2001.

[9] M. Hoffman and P. R. Cook, "Real-time feature-based synthesis for live musical performance," in *Proceedings of New Interfaces for Musical Expression (NIME)*, New York, 2007.

[10] T. H. Park, Z. Li, and J. Biguenet, "Not just more FMS: Taking it to the next level," in *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, 2008.

[11] W. Hsu, "Using timbre in a computer-based improvisation system," in *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, 2005, pp. 777–80.

[12] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: O'Reilly Media, 2008.

[13] R. Dannenberg, B. Thom, and D. Watson, "A machine learning approach to musical style recognition," in *Proceedings of the International Computer Music Conference (ICMC)*, Thessaloniki, Greece, September 1997, pp. 344–347.

[14] C. Joder, S. Essid, and G. Richard, "Temporal integration for audio classication with application to musical instrument classication," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 1, pp. 174–186, 2009.

[15] C. Thornton, *Techniques in Computational Learning: An Introduction*. London: Chapman & Hall, 1992.

[16] A. Hutchinson, *Algorithmic learning*. Oxford: Clarendon Press, 1994.

[17] A. Hazan, R. Marxer, P. Brossier, H. Purwins, P. Herrera, and X. Serra, "What/when causal expectation modelling applied to audio signals," *Connection Science*, vol. 21, no. 2-3, pp. 119–143, 2009.

[18] N. Collins, "A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions," in *AES Convention 118*, Barcelona, May 2005.

[19] F. Gouyon, "A computational approach to rhythm description: Audio features for the computation of rhythm periodicity features and their use in tempo induction and music content processing," Ph.D. dissertation, Universitat Pompeu Fabra, 2005.

[20] P. R. Cook, *Real Sound Synthesis for Interactive Applications*. Wellesley, MA: AK Peters, 2002.

[21] D. Stowell and P. M. D., "Adaptive whitening for improved real-time audio onset detection," in *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, 2007.

[22] C. Kiefer, N. Collins, and G. Fitzpatrick, "HCI methodology for evaluating musical controllers: A case study," in *Proceedings of New Interfaces for Musical Expression (NIME)*, Genoa, Italy, June 2008.

[23] B. Hsu and M. Sosnick, "Evaluating interactive music systems: An HCI approach," in *Proceedings of New Interfaces for Musical Expression (NIME)*, Pittsburgh PA, 2009.

[24] D. Stowell, A. Robertson, N. Bryan-Kinns, and M. D. Plumbley, "Evaluation of live human-computer music-making: quantitative and qualitative approaches," *International Journal of Human-Computer Studies*, vol. 67, no. 11, pp. 960–975, 2009.

[25] S. Hayden and M. Kanno, "Towards musical interaction: Sam Hayden's Schismatics for e-violin and computer (2007, rev. 2010)," in *Proceedings of the International Computer Music Conference (ICMC)*, 2011.

[26] M. Boden, *The Creative Mind: Myths and Mechanisms (2nd edition)*. New York, NY: Routledge, 2003.