NIGEL MORGAN AND NAOMI OSUGA

TOWARDS A CONSTRAINT-BASED SYSTEM FOR INTERACTIVE PERFORMANCE AND RECORDING

Nigel Morgan has been visiting Director of the Electroacoustic Studios at Bretton College since 1998 having held similar posts at Anglia University Polytechnic, Dartington College and Liverpool Hope University College. He is a freelance composer, performer and software developer. He is one of the few performer / composers in the UK specialising in interactive computer music and his works are widely performed and broadcast in the UK and abroad. Noami Osuga studied Contemporary Piano Techniques at CalArts, USA. before coming to Bretton to take a postgraduate degree in Contemporary Performance Arts. In 1999 she was awarded an Alec Clegg Bursary to enable her to return to Bretton to work as a research assistant with Nigel Morgan.

[Published in The National Arts Education Archive Occasional Papers in the Arts and Education. Volume 9. ISBN 1 900 857 707]

Introduction

Over the past twenty years many performers in contemporary art music have been tantalised by the possibilities of technology providing a virtual extension to instrumental capability and human virtuosity. MIDI, a data protocol able to record and transmit performance information to digital samplers and synthesisers, already makes possible a vast extension of the sonic and musical potential of an acoustic instrument: a violin may 'play' the sounds a Gamelan orchestra, 'speak' the text of a poem, generate accompaniments, add ornaments and figurations, even control lighting and other digital media. Despite this phenomenon few performers have been able to harness such technological possibility to their performance practice. This is possibly because the interface for such work is not only highly complex and disparate but invariably separates the sound object from the musical event[1]. Performers are faced with an 'embarrassment of riches' which tends to disable creative progression beyond the 'one-off' composition.

Constraint Techniques is a relatively new field developing in computer science and AI designed to encourage the creation of simple, powerful yet extendible interfaces. A musical instrument is, in part, an example of a constraint system: its very restrictions enable composers and performers to collaboratively realise their musical intentions. In the process of studying how a musician can work effectively with new interactive media the authors have designed and implemented a prototype to support the performance and recording activities of keyboard performers. This paper explains the background to and rationale of this developing system and how it is able to support the capture of ephemeral musical and gestural interaction between the performer, the instrument and the technological interface.

Our paper describes in two parts the evolution of a body of research in contemporary music performance practice instigated by a postgraduate student and facilitated by a visiting lecturer. In Part 1 the process of examining the existing tools and practice of the interactive performer / composer is undertaken to prepare a short recital concluding the student's MA programme. Part 2 describes a short period of further study and research using Constraint Techniques to design and implement a transportable interface (i.e. able to work across different software and hardware systems) for interactive performance.

Part 1

The starting point in June 1999 focused on the preparation of a recital to support the student's interest and preference for electroacoustic music produced by real-time interaction of a live performer with computers. Although the technology behind such interaction has been in the public domain for some fifteen years few performers have developed either the repertoire or the performance practice to bring this unique medium to the concert platform: it is predominantly the stuff of the specialist conference or the research centre. Performers and composers who embrace live collaborations with the electroacoustic medium have favoured the use of studio-made pre-recorded material. The work that does exist in the interactive medium, and is available to the artist, is tiny in comparison and rarely performed.

For both student and lecturer this project provided a unique challenge, no less because the student had no previous experience in the music technology or software employed in the interactive medium nor any familiarity with the existing repertoire. The lecturer was faced with a mere 12 hours of tutorial contact to conjure up a programme that like Bach's *Well Tempered Clavier* might combine pedagogic revelation with artistic integrity and purpose.

Behind the student's preference for the interactive medium, a preference developed as a result of contact with the composer David Rosenboom during undergraduate studies at CalArts, was the notion that this was a possible pathway towards developing a new virtuosity in keyboard performance.[2]

The recital, in effect, presented a short journey through a number of software solutions to interactive performance, the student experiencing just enough of their individual mechanisms to develop an overview of their potential and their short-comings.

But what exactly is the nature and effect of such real-time interaction in keyboard performance? Listening to a performance by a pianist an audience comes armed with prior experiences and assumptions about the quantity and quality of the mechanical touch of 243 strings. A piano is a uniquely expressive instrument in that its performance interface, the keyboard, presents a visual paradigm for the pitch continuum. The performer's gestures invariably articulate motion, dynamics, energy, and the structure of note and sonic events. The sequential movement of fingers produce scale and figurated passages, the simultaneous attack of more than one finger results in chords. As to the sonic content, tone is an amalgam of gradations of the intensity of touch, displacements of register, and the control of articulation of durations between and among note events. Acoustically the instrument is relatively stable, it cannot produce inflections of intonation, pitch variation or vibrato - unlike its early predecessor the clavichord. The piano's lack of timbral variety and instability is compensated by its ability to produce harmonic simultaneity and harp-like resonance. It was only with the remarkable prepared piano pieces of John Cage that the piano as an instrument discovered both its rich percussive and timbral potential.

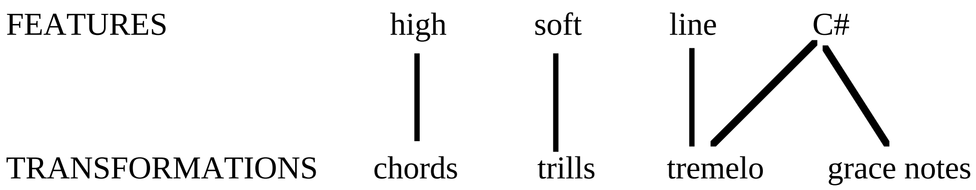
But in the interactive world of keyboard performance the computer can turn every pianistic convention on its head: the pitch continuum can be completely reversed, inverted, or reorganised. Totally new pitch configurations, known in interactive media jargon as key groups, can be constructed. Touch, described as finger velocity and extended in the MIDI protocol by aftertouch (holding the finger down after the initial touch and applying pressure - as a clavichord player would to achieve vibrato), can become a means of adding vibrato, pitch inflection, registral change, and timbral cross fading or filtering. The whole keyboard can be made unstable, reactive: single notes can generate chord complexes in registral configurations unreachable by the fingers, accents can trigger complex arpeggios derived from pitches accumulated in previously performed time frames. The computer acts as a listening machine, listening intelligently, parsing what was previously ephemeral performance data into musically structured responses and additions.

Such a complexity of possibility is the first stumbling block of the interactive art. Not only is the performer able to harness, through digital sampling, every recordable sound in the acoustic and electronic soundscape, but bring an infinite number of conjunctions of action and reaction together; anything that can be recorded can be the source of interaction. The favourite adage of those pedalling the latest software innovation - that your imagination is the only boundary to creativity - has some vestige of truth within it. What is difficult is to achieve is the taming and constraining of this embarrassment of riches into a language state that the performer and the composer can share, and that is cognitively comprehensible to the listener. Interactive music is littered with one-off experiments: fun to develop, challenging to play, interesting to listen to, rarely to surface after the first performance, going nowhere.

Of the major figures in contemporary music probably only Boulez in his ground-breaking score *Repons* of the early 1980s has successfully conceived an interactive composition that has survived critical and analytical attention [3]. Tod Machover has produced a series of cogent, progressive pieces in his Hyperstring Trilogy that extend the sound and technique of acoustic stringed instruments in remarkable ways. One notable example is Machover's solo cello composition for Yo Yo Ma[4]. But the technological backdrop for such pieces is formidable indeed and provides little encouragement or example for the solo performer without the resources of research centres such as IRCAM or MIT.

In searching for an accessible model of interactive practice, innovation and compositional progression beyond the one-off experiment, the research and music of Robert Rowe stands out. His elegant and minimal software Cypher was recognised by the lecturer facilitating this recital project as probably the only interactive software ideally suited to programming by a performer lacking any prior experience in the interactive medium. Rowe is also one of the few composer / programmers to have written with authority about the interactive medium.[5]

With Cypher the computer listens simultaneously for six different features of the performer's musical execution: register, speed, dynamics, duration, harmony, density. Imagine a single event consisting of high, fast, loud, staccato, C Major chords. Cypher analyses and classifies this input, and then the performer may configure a response by applying selected transformations through patching a feature or features to a transformation or transformations.



In the above example high notes will produce chords, soft dynamics will create trills, a single line texture (melody) generate tremolos and the note C# a flurry of grace notes <u>and</u> tremolo.

Cypher's interface is a model of simplicity employing a patching system commonly found in the sound studio where an audio signal from a synthesiser might be 'patched' by a cable to the input of a mixer. In a matter of seconds a 'patch' can be drawn up on the screen and interactions become 'live' instantaneously.

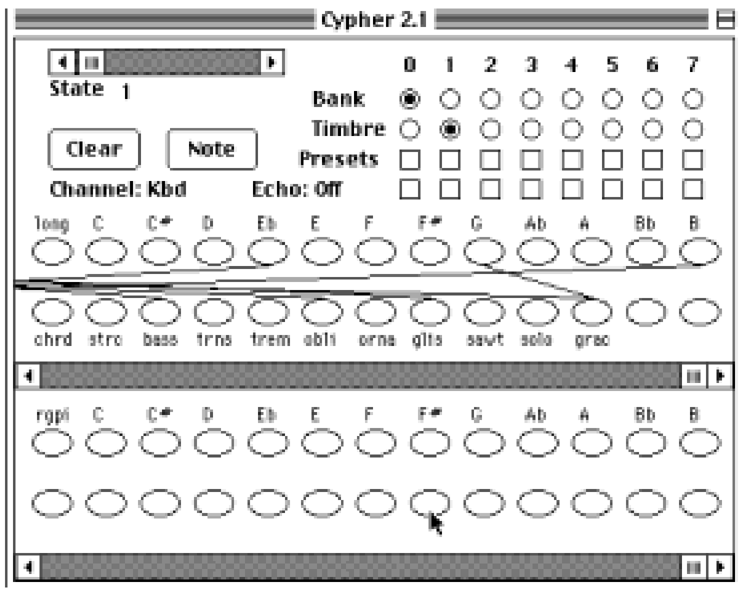


Figure 1. A Cypher 'state': the right hand portion of the interface showing the opening 'state' for Nigel Morgan's *Interaction-2*.

Rowe has given careful thought to his taxonomy and arrangement of listening and playing modes. The system allows for an extraordinary complexity of data flow and interaction but also offers an accessible and easily learnable way into its use. It is important to stress that the musical event interaction component of Cypher is geared to the performer rather than the composer. The performer needs to act quickly, intuitively, on-line rather than off-line. The composer is used to the off-line, the reflective, and is more likely to have programming expertise. Regrettably, two elements of Cypher lie beyond the skill and patience of most performers: one is the matrix programming of instructions for timbral change, the other is the devising of conditions to be met by the player to operate automated sequential movement from one 'state' (a collection of patches) to another.

In the student's recital programme Cypher was used as the interactive medium for an extended work in five movements specially composed by the lecturer. Unlike Rowe's own work *Banff Sketches* [6] for solo piano, Nigel Morgan's *Interactions* [7] for piano left hand employed a single but highly complex interactive 'state' for each movement. This simplified the devising process between composer and performer making it possible to gradually assemble an interactive state so that the performer could test each listening feature and each performance transformation to find exact tolerance of action and machine response.

To achieve a measure of critical comparison between interactive systems a second software package was identified as offering a very different interactive experience. The 'M' software [8] created by David Zicarelli is the precursor of MAX [9], an object-oriented programming language of considerable complexity now recognised with its digital audio counterpart MSP[10] as the *de facto* standard for the origination of interactive systems. In comparison to Cypher, 'M' is predominantly about recording, triggering and controlling pitch input. The system records melodic phrases or chords without capturing any kind of rhythmic and dynamic material associated with it. The performer can then interact with this series of notes by applying transformations of register, speed, dynamics, note-order and note density, timbre, new rhythm ordering, articulation, all in real-time.

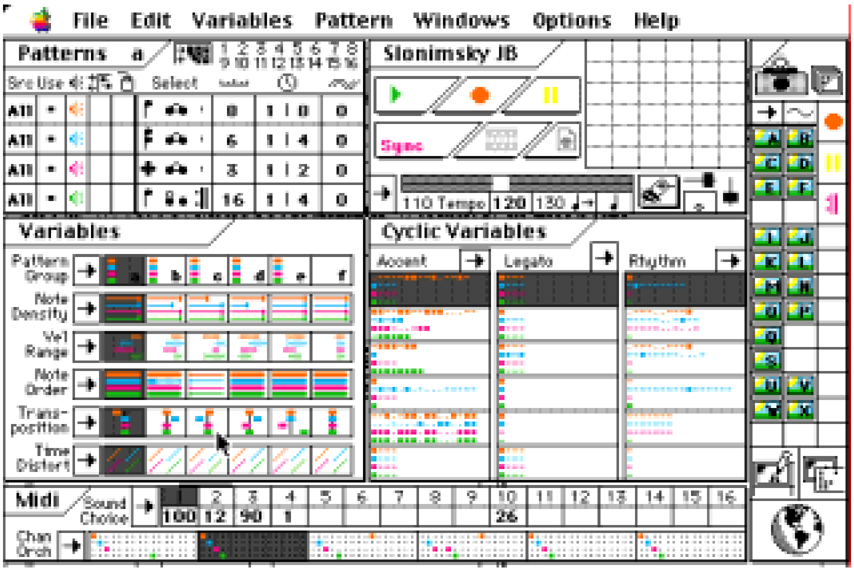


Figure 2. The screen interface for David Zicarelli's M software showing the entire program for Nigel Morgan's composition *Slonimsky Jambox* for solo performer with MIDI technology.

In the Technical Report[11] on her recital preparation the student writes about the 'M'software:

> For me M is about making patterns and controlling structures at either note level (note order or density) or at structural level (controlling different patterns by triggering 'snapshots' of combinations of transformations).
>
> M also enabled me to use different modes of performance gesture: using the computer keyboard, the movement of the computer mouse and the MIDI keyboard through one and two stage key trigger combinations. All interactions and changes of pattern and structure can be captured in a M 'Movie' which is saved in the universal MIDIfile format. This data recording enables me to review and edit every nuance of performance detail in a MIDI sequencer or scorewriter.

Like Cypher the important factor for the performer is that the interface is consistent, the tools for interaction are already in place, and a basic technique for its use can be acquired in a few hours. Unlike Cypher it has been widely used as a composing, improvising and performing tool.[12]

But both systems focus on the control and manipulation of the musical event rather than sculpting of the sonic object. This dichotomy between event and sound has been highlighted by several composers: by Ole Parmerud whose view is predominantly positive[13] ; by Leigh Landy [14] who rightly trashes the MIDI protocol that engenders this split whilst intriguingly employing MIDI interaction to great effect in his performance work with Jos Zwaanenburg [15]; somewhere in between this Miller Puckette looks forward to 'a life after MIDI' when 'we can treat pitch, rhythm and timbre as a continua'.[16] Whatever one's position on the MIDI protocol that lies behind the systems described above, it remains after some fifteen years the only viable control interface for the 'rest of us'. There follows a description of MIDI for the uninitiated taken from a paper by the lecturer at the 1999 CTI Colloquium 'Beyond Art?':

> The Musical Instrument Digital Interface is a data protocol that deals not with sound itself but the parametric ingredients of music - for example, pitch, duration, dynamics. Remember you can't hear the pitch of middle C unless something or somebody plays or sings it. A MIDI keyboard makes no sound; it simply sends performance information - the keys I choose to press, the various durations my fingers decide to hold, and the individual velocity of the attack each finger brings to a key. The MIDI keyboard sends all this in a serial stream of data through a computer to a sound card or synthesiser which interprets this data to trigger appropriate sounds. My performance data (not the sounds that ensue) can be recorded and played back using software known as a sequencer. Sequencers are able to capture such musical performance information and most can display it on screen in musical notation in varying degrees of accuracy and interpretation.
>
> From its inception the higher echelons of the digital music fraternity have been looking down their noses at MIDI as a technological dinosaur that has to disappear. Yet it hasn't disappeared. It continues to thrive because it can handle parametric information such as pitch, duration and dynamics in a powerful and efficient way. And although we have the ability to effect incredible transformations on a soundfile of acoustical data we can hardly begin to separate the basic parametric information indicated on a musical score and readily and effectively captured in MIDI performance data.[17]

The remaining component of the student's recital was the use of the MAX software, not as a programming language from which personal interactions could be designed - a long and difficult task for the novice - but as a completed MAX application. Karlheinz Essl's remarkable *Lexikon Sonata* [18] is an interactive composition for MIDI-driven solo piano which contains a library of specially designed objects that capture the generative structure but not the discrete content of complex musical events.

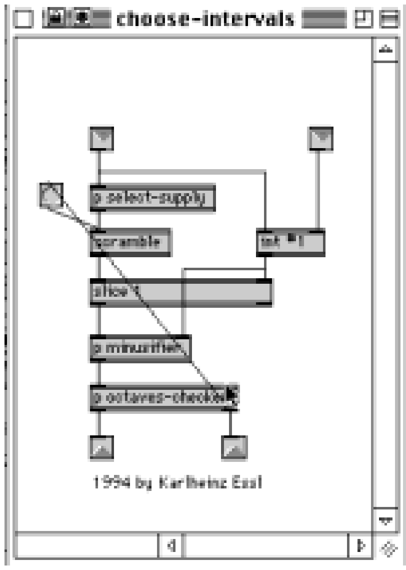


Figure 3: a MAX Patch for choosing intervals from an input stream of MIDI data.(From Karlheinz Essl's Real-Time Composition Library)

Investigation of the generative structure of musical events is the essential background component of algorithmic composition and performance. All three software packages employ algorithms able to generate musical events from predetermined real-time structures. In order to construct effective algorithms the designer invariably makes use of the most fundamental device offered to us by computer technology, the random number generator. No better or clearer description of the power of this device has been given than by Hiller and Isaacson [19] who were amongst the first composers to utilise the computer as a tool for real creativity rather than as medium for recording and editing in their collaborative composition the *Illiac* quartet.

The random number generator along with generators based on mathematical representations of chaotic systems, such as Marchov chains, can be made to simulate the kind of low-level decision-making processes composers, improvisers and performers engage in. The lecturer's own research into reflection about intention in musical composition has time and again revealed the very arbitrary nature of a human composer's choices of parametric events.[20]

Thus, in performing with an interactive system the musician is engaged with responses from machine listening and algorithmic processing often directly based on studies of musical cognition that go back fifty years. Robert Rowe summarises a rationale for this in the conclusion to his key note address from the Edinburgh International Congress in Music & Artificial Intelligence:

> I believe that there are compelling musical reasons to emulate human musicianship with computers. Rather than a diverting science fair project or invidious scheme to deny performers livelihood, machine musicianship can strengthen and extend human ways of making music . . . the good news is that interactive systems have musical worth even when realising only a small fraction of human musicianship. Using techniques from artificial intelligence and whatever other sources seem appropriate, their worth can be enhanced step by step as our understanding of music cognition deepens.[21]

Part 2

The majority of performers in the field of interactive music media are themselves composers and music technologists. In the UK the violinist Kaffe Matthews and trumpeter Jonathan Impett have over the last ten years developed highly original but startlingly different approaches to their music making. Both share the vision of creating for themselves a *modus operandi* that enables them to bring together composition with improvisation, off-line sound design with on-line signal processing, instrumental performance with realtime interaction by computers. Both musicians are now moving away from MIDI control of samplers, synthesisers and processors to live sampling and real-time digital audio processing. In Kaffe Matthew's work this live sampling is focused at the performance environment itself which provides the sonic material for performance interaction.[22]

At the outset of the second phase of study and research it seemed essential for the research student to qualify her position in this triangulation of performer - composer/improviser - music technologist that is, by and large, the norm of the interactive music medium. By training and inclination the research student is strictly a performer for whom improvisation has been a necessary adjunct to studies of the contemporary piano repertoire rather than an integral part of her musical *persona*. For her it seemed that instead of devising her own musical material she would need to seek out collaborations with composers and music technologists.

Before such collaborations might occur it was felt necessary to declare and establish the components of the musical, sonic and technical resources and possibilities she might require to communicate to composers. It was at this point that lecturer and research student decided to adopt a Constraint Techniques approach to devise a system that would integrate a set of sonic extensions of the piano with a set of musical interactions. To be musically viable in performance these sets of extensions and interactions had to be triggered accurately and instantly from a foot pedal, in order to leave both hands completely free. Furthermore, it was felt that this system needed to be transportable across different hardware and software.

Constraint technology is lively field between computer science and artificial intelligence that makes it possible to declaritively state and solve problems either in numeric or symbolic domains. For artistic applications constraint techniques are widely recognised as a key concept to develop high-level authoring systems, but systems that can be used by people with limited or no computer science backgrounds.[23] In music cognition constraint techniques have formed a lively part of the work of Lerdahl and Jackendoff particularly in developing ideas about musical grammar and tackling the problem of the gap that appears to exist between compositional systems devised by composers and the cognized result i.e. what the listener perceives and understands. [24]

In some respects the collaboration of student and lecturer could be seen as embodying the action research model where the lecturer, who has experience in constraint techniques as a programmer,[25] acts as the change agent. This was achieved by helping the student apply specific techniques to establish a grammar of sonic and musical interactions: a limited set of rules able to generate indefinitely large sets of structural descriptions.

It has been mentioned previously that there exists in the MIDI domain a separation between the control of sound and the control of the musical event. Indeed the protocol for real-time sonic control and interaction is certainly limited, for example it is extremely difficult to achieve sonic transformations from one state to another. MIDI's own data resolution, the data bandwidth able to control change, is very low. However, in this instance these limitations where perceived to be favourable for the task in hand.

In the architecture of MIDI hardware and software synthesisers there are two predominant approaches to sound design. One approach enables a new instrumental timbre to be created and stored as a discrete new instrument, named then aligned to a MIDI program change number which when presented as an in-coming message selects the new timbre. The other approach is to have a chain of editing processes independent of, but able to act upon, a source sound. This means that the source sound, usually a wavetable sample, may be changed but its edited parameters remain in place. Despite the large range of current sound synthesis and sampling systems there is core group of sound editing parameters that the majority of these systems share.

The decision to adopt the second approach moved the selection mechanism away from MIDI program change numbers to the switching of channels. The MIDI protocol allows for 16 separate channels of performance data to flow simultaneously. However, the MIDI keyboard is designed to send its information on a single channel, after all it is usually playing a single instrumental timbre. Student and lecturer recognised very early on in their work together that channel structure and control was the key concept to grasp and secure. The MIDI protocol has no single real-time message that enables a performer to change channel as he or she might change program. The only way it can happen is to place a processing device in software (built in MAX) or hardware (a Yamaha MEP4) between the keyboard and the sound synthesiser or sampler. In this way channel information can be redirected, doubled, trebled, quadrupled. Such channel reconfiguration from a single source channel was found to be possible to implement on both dedicated hardware devices and within interactive software systems, both able to be controlled from messages sent from a foot pedal.

The next stage was to devise 16 different editing configurations, one per channel part, having a clearly recognisable identity and potential for interaction through combination and rearrangement. In our system these configurations are arranged around articulation and inflection, pitch modulation, envelope shape, filter shaping of harmonics, with slots for percussive sounds and monophonic performance with portamento.

The end result implemented and tested in a specially written composition *Array* for solo keyboard performer represents a striking example of the possibilities of sonic extension through real-time interaction. The first movement of *Array* addresses the concept of 'compass' - both in its double meaning of direction and high and low tessitura of instrumental register. In performance the musical result is engaging as the movement from one sonic extension or group of extensions appears seamless because timbral change is being effected through multi-channel reconfiguration rather than re-selection of individual instrument programs on a single channel. The latter tends to chop the natural decay or release envelope of the program being changed, denying any natural overlap to occur into the next sound.

Having solved the problem of defining a common set of sonic extensions accessible in real-time we now move on to defining a similar set of interactions of and with musical events. This appears infinitely more difficult partly because the performer ideally needs these interactions to be as instantly accessible and controllable as the sonic extensions. Many processes one can imagine, such as recording a given phrase and instantly playing it back as a repeating loop require a sequence of commands: to start recording, to stop recording, to start playback, to stop playback.

At the present time the design of this part of the interface is looking at a means of easily aligning chosen event interactions with particular sonic extensions to enable a simultaneity of control, possibly via a patching table that can be applied prior to performance.

As a preliminary step we have chosen to implement a set of sixteen interactions. The five chosen below demonstrate some of the primary characteristics:

> duplicating the note input date to produce between two and four note layers which might be transformed variously by transposition, reversal, inversion;
>
> duplicating note input as a result of setting up conditions governing the velocity of touch to add or subtract further note layers;
>
> transforming the velocity of touch into note data, i.e. repeated attacks on a single key will produce a sequence of differently pitched notes. In the MIDI protocol the velocity range of 0-127 is equal to a nine and half octave pitch compass;
>
> transforming the velocity of touch into the speed of the attack portion of the sound extensions envelope, i.e. when playing a standard piano timbre *pianissimo* notes would be produced without the hard attack noise transient, the resulting sound would appear rise from beneath the player's fingers; when *fortissimo* , only the initial noise element of the hammer action hitting a metal string might be present;
>
> transforming the pitch difference between each note to the interval of a quarter tone. In chordal playing this would make possible wholely new harmonic relationships.

The authors feel such sonic extensions and musical interactions as described above present a viable framework from which new music and new musical relationships can be evolved. With these formal extensions to musical grammar the performer can extend the very language of piano playing. And, to make this language accessible the system is seen to be transportable in a way that would enable a collaborating composer to get his or her hands on these new extensions and interactions. It is the student's intention during the next phase of her research to look at implementation of these components as a downloadable application on a personal website, software that could be configured directly to a computer's soundcard or MIDI keyboard.

In using this technological interface the element of recording is intrinsically different from the norm: it does not capture what the listener hears but what the performer actually plays, partly as gestural action, partly as a 'movie' of the constantly changing state of the control of sounds. This means the recording does not contain any acoustical data only data governing the actions that make sounds happen. This is not an entirely recent phenomenon: in analogue technology many of the great pianist / composers during the first 30 years of the 20C punched piano rolls that recorded every nuance of rubato and touch. A piano roll contains no sound, just the instructions to play sound on an appropriate player-piano.

The MIDI data stream that evolves from keyboard touch and the intervention of extensions and interactions can be continuously captured by the software sequencer. As described in Part 1 the sequencer is a recorder and editor of MIDI data able to record musical performance (but not the acoustic sound) at a very high resolution and display it in either clock or virtual time (bars and beats). Its use as a tool for studying performance practice, particularly rubato, is well established.[26] But for many composers the search for the right notes that proceeds from the moment of inspiration begins at the keyboard. Indeed, for composers such as Stravinsky, inspiration could only 'happen' when he was able to touch the sound. To capture this 'search' in real-time and be able then from the recording to analyse, edit, amplify, extend and ultimately reduce this material into the building blocks of composition has now become an important working mode for many musicians. There is slowly developing a known body of art music that is the direct result of the improvisation - recording - editing paradigm, notably in recent works by composers such as John Adams, Dominic Muldowney, and Michael Nyman.

It is useful to remember that since the evolution of audio recording there has been a move away from regarding the keyboard as the source from which the listener begins his or her relationship with art music. Until the middle of the 20C new music, particularly orchestral scores, were regularly transcribed for hands at the piano. Beethoven transcribed all his symphonies for piano trio, Webern transcribed Schoenberg's Chamber Symphony No.1 for piano quartet. Today, sadly, such is the complexity of musical utterance, many composers view the electronic experience of the CD recording as infinitely preferable to listeners 'getting their hands on the music'.[27]

The technological extension of the keyboard and its performance practice may offer one way forward to composers , improvisers and performers to rediscover the potential of the keyboard not just as a medium for performance but as a means of creating and realising new sounds and new music, taking a few steps beyond what is physically possible with two hands, ten fingers and 243 strings.

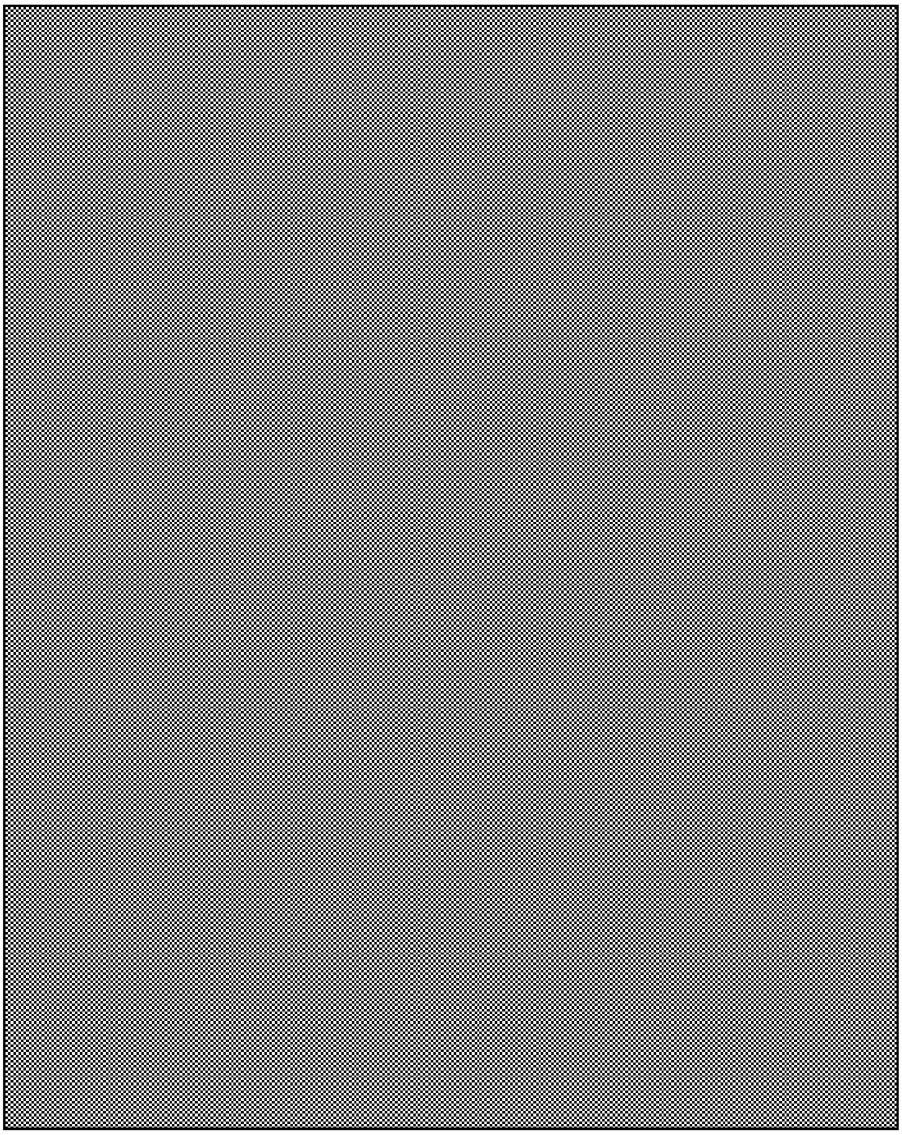


Figure 4: An MSP patch for ring modulation of an acoustic piano from a vocal source designed for Naomi Osuga by Dave Moore and Braham Hughes at Barnsley College