THE HANDS, a set of remote MIDI-controllers

MICHEL WAISVISZ

STEIM
Groenburgwal 25
1011 HR Amsterdam

1. INTRODUCTION

Even the most recently developed electronic music keyboards still provide rather poor translation of the rich information generated by hand/arm gestures and finger movements/pressures. Experiments undertaken during the last 3 years at STEIM have shown that a multi-sensor control device seems to be the appropriate approach. The Hands thus were designed to include distance/speed and tilt sensing as well as a set of multi-function switches and potentiometers.

The Hands consist of three parts: the two "hands" and an analog-to-MIDI-converter. The "hands" are a set of two aluminum ergonomicly shaped plates with sensors, potentiometers and switches strapped under the hands of the performer. The analog information generated by finger movements and changes in the positions of the hands are scanned by a microprocessor with an onboard MIDI encoding program. The MIDI codes can be sent to any MIDI device through a standard MIDI cable. The Hands can be easily carried in a small bag (40*20*20 cm).

Since the premiere of "Beat Concert" at the Amsterdam Concertgebouw in June '84, many concerts have been given with a prototype of the Hands. An important experience gained from playing with the hands was of possibilities for control signal manipulation. Rescaling, patching, and other treatments, i.e. the application of a control-signal algorithm,are the main factors through which a controlling device derives its effectiveness. For this reason later in this paper a fair amount of attention will be given to the concept of a control-signal processor, a system that can easily go beyond the possibilities of MIDI and will bring VOSIM and Chant-like systems under real-time control. It will become clear that the present state of the Hands -mainly gesture sensing, simple finger control and almost no control-signal processing-, is certainly not the final one, but the potential qualities can be seen and heard.

2. RELEVANT BACKGROUND INFORMATION

My activities in building electronic music instruments started in the late sixties. I was not quite the type of composer to spend 3 hours in a studio to get 6 seconds of passable music, I wanted to hear results fast in order to make the right decisions within the same train of musical thoughts. Of course I did spend many hours in the electronic music studio. The first instrument built was a hand controlled tape-loop playback system. Seated between two stands equipped with tape playback heads I could move by hand two tapeloops with the same condensed sound sequences. Synchronised use of two foot controlles amplifiers and the movements of the two tape loops provided the possibility of continuous play back of the sounds. Experiments with my fingers on the printboards of the first Moog in Holland (Royal Conservatory of Music, The Hague) and later inside the Putney VCS3 showed me a very direct way of playing these instruments. Soon research started to become more elaborate and STEIM adopted the project. Many "Crackle"-instruments were developed amongst which the Crackle Box, The Crackle Synthesizer and an early polyphonic organ (48 voices i.e. 48 crackle boxes!) were the musical products. For theatrical use, amongst others, a special Crackle Stage and fencing devices were built.
The Crackle instruments were analog electronic music instruments. They were based on a feed back oscillator circuit. Some of the wires in the circuit were diverted to touch pads. The conductivity of the skin would enable the performer to play the instrument by placing the fingers on different groups of touch pads. Patching and controlling was done by respectivly choosing the right pads and applying varying degrees of pressure or by moistening the fingers.

This background information is important in order to underline my belief that these analog instruments are conceptually far superior to many contemporary digital systems, both my own and those developed by others. An aspect of the Crackle instruments I find crucial is that control-signals and audio-signals can easily be mixed; a phenomena that allows the performer to have extremely dynamic control over the behaviour of the circuitry, and that is not yet available with small real-time digital systems. But I went digital because of reproducibility, and the expected sound quality of digital musical instruments. As a composer/performer, I'm working on integrating the advanced features of analog control into the hands. Within a few months I'll be able to write another paper with the Hands instead of about the Hands. I feel that my confidence in digital instruments will increase as soon as we get the advanced features of analog control conceptualised in a new way inside the Hands.

3. TECHNICAL DESCRIPTION

The Hands

The following describes the hardware of the actual prototype. The design of the system was started together with STEIM engineer Johan den Biggelaar, later changes were made by, and further developement will be persued with, Wim Rijnsburger.

In each hand there are three rows of four keys; these provide pitch control (MIDI key-on, key-off) within one octave. Fitted under each hand are four mercury switches aligned to the cardinal directions and defining a conical surface. These switches control "octave-transpose". The conical positioning allows 1 or 2 switches to be "on" at the same time if the hand is tilted in this conical plane (like caressing the upperhalf of a globe). The 8 possible combinations correspond to an 8-octave range.

In the prototype both hands control the same "key-board"; each hand can play from low to high, and back, without crashing into the other. The left hand contains a sonar transmitter aimed at the right hand. The right hand has a matching sonar receiver. The actual distance between the two hands is measured by comparing the delay of an ultrasonic pulse between the two hands with the original pulse time. This data is assigned to the key-velocity value. With proper scaling one has, compared with a "velocity sensitive" keyboard, a very high resolution effect over the entire range of hand positions, from very close to fully outstreched. One should note the extensive timbral implications of this and, especially, the great amount of control obtained when this key-velocity value is mapped to separate oscillators (DX7:operators) in a FM sound-generation environment.

The left hand provides press-buttons for the selection of 3 different MIDI channels (1, 2 & 3) or all three channels simultaneously. The channel assignment is for both hands. The right hand has push-buttons for program choice (DX7: voice 1 to 32). One steps up through the programs, the other down. A reset (to voice "1") has been removed.

A thumb-wheel potentiometer is connected with the pitch bend function an a push-button is available to toggle sustain on and off. Used with the DX-7 "step"-parameter set to "1", appropriate use of the pitch-bend thumb-wheel makes it possible to play chords which otherwise would be impossible to "grab" (because of the octave-"window") or would, uneconomically, have to be played with the two hands in different octaves.

Another push-button toggles the "scratch"-function on/off. If "on", it will cause a new key-on event, for every note still "down", at every change perceived by the sonar detector. This is one of the more interesting possibilities of control with the hands. Continuously controllable timbral changes become possible as a consequence not only of the changing velocity parameter but also due to interference between the rate at which the "scratch" function generates updates and the attack times of the DX7 voices. Programming the voice-parameters especially for this "scratch" function one takes a group of voices normally rather eerie and makes them expressive under "scratch" control, sometimes, when sustain is used, extremely dynamic-visually, organically connected with its audible appearance. By moving one's arms while having keys pressed one "bows" the sounds.

The circuitry for the sonar-transducers, the pitch bender, the "pull-up" resistors and the diode-matrix all are mounted on the bottom side of the hands. The power is supplied by the analog-to-Midi converter.

The Analog-to-MIDI Converter

The entire circuitry for the analog-to-midi controller is designed as a single-board micro-processor. It consists of a single chip 8-bit microprocessor (Rockwell 6511), 3 EPROMS containing the program-code, a crystal for the system clock, the MIDI-interface and the power supply. There is sufficient RAM on the 6511 to ru the programs. There are four 8-bit ports on the 6511, one of which is software programmed as a serial port for the MIDI-interface. (fig. 1).

The Software

The program was written in 6502 assembly language on an Apple IIe under the ProDOS operating system. The program was loaded into EPROM from an EPROM-programmer in one of the Apple slots. Trial-and-error was the method since no monitor was available for the single-board processor. At this moment the program is undergoing a total revision. The original program (still used in the prototype) was conceived in a hectic three month period preceding the premiere in '84, and later a lot of "definitively-the-last" changes were added!

The plan of the current program is quite simple: After initialization of RAM and conversion parameters, an endless loop is started that sequentially scans the key matrix and mercury switches, reads the pitch bend position and measures the sonar distance. These data are mapped through tables to the appropriate MIDI command sequences. It was found that the processor and program are fast enough to provide the data for MIDI-systems. Actually most MIDI systems stall when the "scratch" function is used excessivly; "MIDI buffer overflow" messages were easily obtained on the DX7.

4. CURRENT RESEARCH

HARDWARE

Since The Hands are intensively under development at the moment of this publication, a look at the near future of the project is essential. What follows is a list of "Hand" topics on which

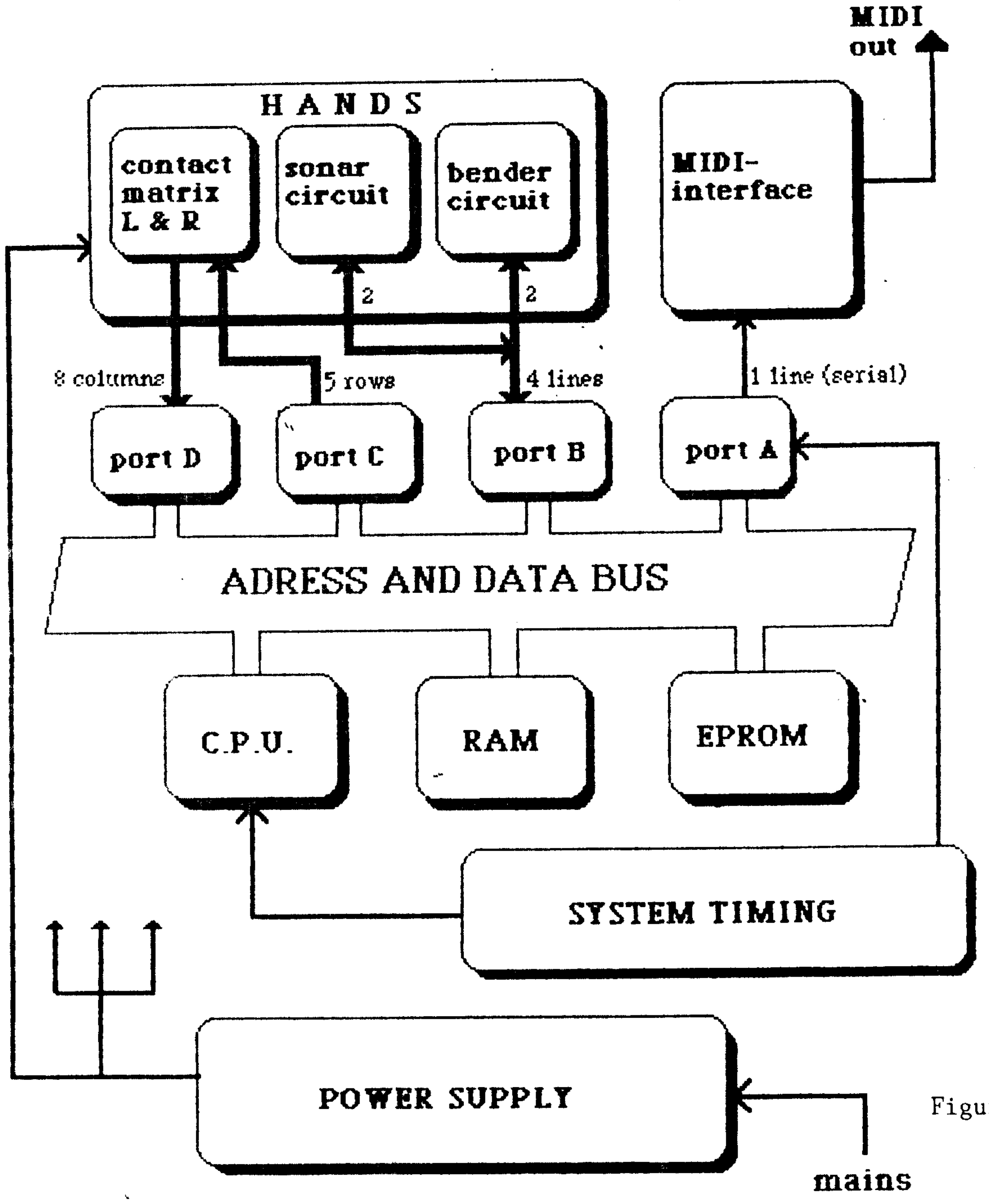

Figure 1

research is being done. After that I will describe the concept of the Control Signal Processor which is being evaluated at the moment at STEIM in collaboration with composer/ software-engineer Joel Ryan.

Wrist Attachment
The actual (prototype) Hands are strapped with elastic bands to the palms of the performer's hands. Without being too tight, the Hands are fixed well enough to keep in firm contact with one's hands during fast and rough tilting movements. Since the Hands fit tightly to the palms, finger movements are strictly finger movements; no weight from the palm or wrist is added to a finger "strike". Experiments are being done with Hands that are attached at the wrist so that "free" movements of one'hands are possible. This will allow a more percussive playing-style.

Electrolyte tilt-measuring
Since the freedom to move the wrist in all directions is less than that of the palm, possibilities are being explored to use highly sensitive electrolytic tilt measuring devices to replace the mercury switches now used for "octave-transpose".

Fiber-Optic Sensor
At STEIM a research-project by Peter Dessain on the possibilities of using fiber-optic sensors for the detection of finger-, hand- and arm-joint movements has begun. In the Hands these sensors will allow fingers to switch and control functions without actually touching keys. This addition will be used in configuration where the hands communicate with a composing computer-music system. By bending one or more fingers sharply at the same time, one will be able to trigger special modes, or to load and activate other programs.

MIDI-Addressing
Each hand will have its separate MIDI-addressing option. A single button in the left hand will replace the four discrete channel addressing buttons. When held down, all "pitch" keys for each hand will become "channel selection" keys. It will be possible to toggle up to 12 different MIDI-channels for each hand separatly. Keeping track of the channels and available voice-programs will definitely increase the need for a visual feedback display.

Program Selection
An extra push-button on the right hand will assign the "sonar-value" to the program-selection number. When it is held down, one can roughly select one of the 32 DX-7 voices by moving ones arms in and out. "Fine" selection is still possible with the single-step selectors.

Extra Sonar Transmitters
Two extra sonar-transmitters will be mounted at the bottom of the hands, aiming downwards. Two matched sonar-receivers, to be clipped to one's trousers, will enable vertical detection of separate hand movements. The resulting values will be assigned by the software to MIDI-control functions. In this case we imagine applications ranging from "modulation-depth" to elaborate "scratch" like extensions of the "interference control" program.

piezo electronic sensors
Piezo electric accellerometers will be used as data sources for the GoWi-algorithm (described later). Piezo electric pressure sensors will be added to the "pitch"-keys to generate data for the Control Signal Processor (described later in this chapter). Preprogrammed configurations of this processor will be triggered depending on the applied finger pressure. Piezo electric strips and variable resistance paints might be used to detect pressure changes under one's palms. Together with the accelerometers and the "key"-pressure sensors they would generate a set of values from which inputs can be made for the GoWi-algorithm.

"Crackle" Control
As mentioned before the application of Crackle touch pads might be considered if it turns out to be impossible to effectively feed the analog output signal back to the digital control structure. At this moment we have not fully evaluated the problems which might arise.

"Radio" Control
For visual and safety reasons, we opted for wire connection between the Hands and the MIDI-instruments. It is not impossible that radio-control will be used, if we start building a series of Hands (requests for customized versions have already started to come in).

SOFTWARE

Midi-Patch
The analog-to-MIDI convertor will be extended with extra RAM for the storage of MIDI parameter data. This will make experimental re-patching of MIDI-functions possible with the use of an external terminal (Mac). The memory will be sufficient to store strings of MIDI-commands allowing each "pitch" key to be individually assigned to initiate sequences of events. Self-modification can be part of a sequence, so that,after executing its function once, a key can start the execution of another sequence, and so on.

Extended Interactive Converter
Steim enigeers Aad te Bokkel en Paul Spaanderman are working on a fast, multi-channel parallel MIDI mixer for the Mac. This will enable us to extend the possibilities of interactive MIDI control-signal processing described else where in this paper. The software will be written in Forth. Experiments are also being done with LOGO as a handler of high level compositional block-decisions. If it turns out to be succesful, two Macs will work together: the "composer" in LOGO and the "performer" in Forth and, where needed, in assembly language.

Video Synthesizer Control
In the winter of '86, I will perform a piece where two video-cameras will scan the movements of the Hands during the concert performance. The video signals will be routed to the Fairlight Computer Video Instrument for treatments and mixing. The resulting images will, be projected on two relatively small screens beside me on the stage. The Hands will, through the MIDI-converter, simultaneously provide control inputs for the Video synthesizer and for the sound synthesis instruments.

DX7 Voice Editor
A program in EPROM for the analog-to-midi converter reconfigures all Hand controls and sensors into a programming device for DX7 voice. (The sonar detectors will be the "data-slide"!). The program can be used during performance by switching a <ctrl> function on the Hands.

THE CONTROL SIGNAL PROCESSOR (CSP) (fig.2)

This system is designed to increase the quantity of information obtainable from The Hands, or from manual controllers in general. The reather "simple" information generated by finger movements would be applied to a network of relationships to stimulate a complex change in the multiple outputs of this network. The system will be built around wave or data tables that are read out by a set of pointers with flexible and manually controled scanning patterns. The output can be fed to computer music instruments either via the appropriate interface or to analog electronic music instruments via a digital to analog converter and a Crackle-box-like control system. We expect to have the prototype working in the fall of 1986.

Working with the control-signal-processor (C.S.P.)
1. A personal computer loaded with a C.S.P.-configuration program is connected with the CSP.
2. The 32 wavetables are filled with patterns.
3. Pointer-looping/paths are defined.
4. Patching is set up.
5. These data are stored on floppy disk as "set-up" for a piece or a part of a piece.
6. The system is installed in a performance space. A controller is conneceted as well as the sound synthesis instrument(s).
7. A "set-up" is loaded into the "buffer" memory location.
8. At this moment it is possible to load these data into the "controller" memory. Although the patterns in the wavetables remain fixed, the patching and pointer paths can be adjusted "live".
9. The actual start and modulation of the

pointer-movements is controlled by the finger actions on the controller

Some brief descriptions of the controller/CSP relationship

Pressing a key on the controller can, if the C.S.P. is configured that way, start a series of pointer-movements through the tables. This will generate a sequence of vectors that, through patching, can be assigned to the control inputs of, for instance, Chant or Vosim "generators". The result is a sound sequence. The lenght of the sound sequence is determined by:

- the algorithm of the pointer paths
- the reading of speed of the pointers.

In other words, the depression of a key can, in musical terms, be seen as either the start of a note, or of a melody, or of a sound sequence. Depending on the capacity of the system, it could even be the start of a year long piece. The "controller" memory can be filled in such a way that each key selects a different C.S.P.-configuration. So, under each key is stored a different sound or sequence or piece.

Proportional finger/hand/foot control of sequence "parameters" initiated by a key depression, is possible by using patches where sonar pairs, pressure/bend-sensitive keys/knobs, accelerometers, etc. are connected with pointer modulation functions. Depending on the configuration of the pointer output patching, finger/hand/feet movements can directly control all timbral and time-related aspects of the sound sequence. "Indirect", algorithmic control, is possible by using pointer outputs as inputs for "live"reprogramming of the pointer scanning algorithm. By loading the same "sound" under each key, and augmenting the specific wavetable that controls "pitch" with the desired value for each key, the Controller/C.S.P./synthesis-system can be turned into a conventional electronic music keyboard-synthesizer. Filling the wavetables with "outside world" sound samples changes the C.S.P. itself into an autonomous "synthesis" system.

Control signal processing algorithms

I see the important role for interactive systems in the realisation of "Control Signal Algorithms" As a starting point for further thinking, I will mention five main algorithms that I have conceptualised for this task:

The GoWi (go wild) algorithm detects all control information generated at every moment in the piece. If a certain threshold in the qantity of control information is passed the algorithm generates wild control information by itself and mutilates the original control-signals.

The GoOn (go on) algorithm doesn't change any control information. The "normal" control procedure is performed but tests on the incoming control data are performed and branches to the other four algorithms are executed for particular sets of conditions.

The what algorithm questions:
what did I do?
what did I want?
was it good?
what do I do now?

The T.E. algorithm (Thats Enough) basically decides at rather odd moments that things start to become boring and stops execution of the actual state. It is extremely alert during execution of the GoOn algorithm. The composer/performer is forced to restart actions. If he/she waits too long there can be a call for the last data of the GoWi algorithm. This will be performed as a suggestion.

The Effort algorithm is the one that has my full attention at the moment. It is possible to build any sort of easy controller nowadays. As listeners we are interested in spiritual efforts made by the composer/performer. These spiritual efforts can be seen only through the physical efforts the composer/performer makes during the performance. High controller-ergonomics provide effortless composer/performer actions and uninteresting music is the result. There is a musical need for artificial "friction" in the concept of new controller design. This friction can be mechanical (heavy hands, slow buttons, wide arm-movements) or compositional (spontaneous modification of control-data by the control signal processor,

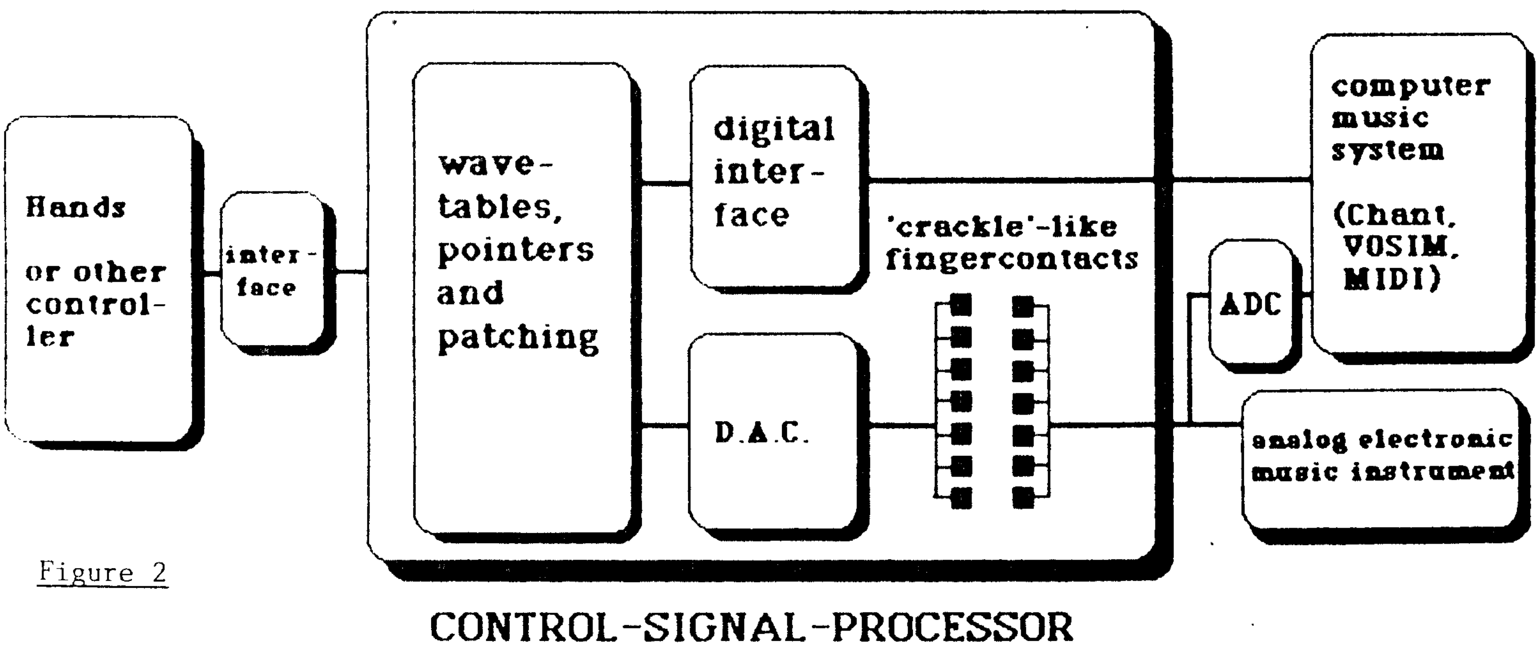

Figure 2

creation of new data by the same device, etc.)

5. CONCLUSIONS

Finally, I would like to get into some generalities about electronic music performance with devices such as the Hands. The Hands provide a vast amount of control over the musical material, but this is far from a quarantee of a good piece. I see live composing as the richest musical experience. The demands on the composer/perfomer are very high; I haven't seen anybody succeed in a reasonable way yet. Until now the lack of adequate controllers has been a considerable obstacle, though many interesting controllers should turn up in the comming years. However, other problems encountered in live composing remain to be solved.

We still cannot listen and play/compose in the same time and keep track of all developments in a piece. Only if the structure of the piece is well defined, as in a lot of jazz-situations, do some composers/performers reach a high level of simultanious listening/playing. By "listening" I mean hearing the developing structure of a piece (a piece not composed in advance in the comfort of the erudite composers residence). Interactive composing systems might help us in the beginning to keep track of some preconceived structure, but I think they are valuable only if one does not rely on them too much during the performance. As long as they surprise the composer/performer while keeping track of the program structure, these systems will be a good learning aid - until we are able to do the job ourselves.