I. INTRODUCTION

Since the recently developed electronic music keyboards will provide richer and more powerful manipulation of the rich information generated by hand-in-hand gestures and finger movements, experiments undertaken during the last 3 years at STEIM have shown that a multi-modal control device seems to be the appropriate approach. The Hands, as they were designed, include distance-sensor 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-MAID converter. The "hands" are a set of two aluminum ergonomically shaped plates with sensors, potentiometers and switches, striped 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 MAID encoding program. The MAID code can be sent to any NIH device through a standard NIH cable. The Hands can be easily carried in a small bag.

Since the premiere of "Beau Concert" at the Amsterdam Concertgebouw in June '74, many concerts have been given with a prototype of the Hands. In a recent experiment gained from playing with the Hands was of possibilities for control signal manipulation, Reshaping, 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 machine that can resolve through the possibilities of MAID and will bring MAID into a CIASS-like system under direct 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 4 seconds of recorded music. I wanted to hear results fast in order to make the right decisions within the same range 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. Sorted between two stages equipped with tape playback heads I could move by hand two horizontal boxes with the same concerned sound sequences. Synchronized use of two foot-controlled amplifiers and the movements of the two tape loops provided the possibility of continuous playback of the sounds. Experiments with my fingers on the printboards of the first Mungo Holland (Royal Conservatory of Music, The Hague) and later Jessesthe Puteaux VEC showed me a very direct way of playing those instruments. Some 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 technical 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 feedback oscillator circuit. Some of the wires in the circuit were clamped to touch pads. The capacitance of the skin would enable the performer to play the instrument by placing the fingers on different groups of touch pads. Tuning and controlling was done by respectively choosing the right pads and applying varying degrees of pressure or by moistening the fingers. This background information is important in order to understand the role of the instruments. The MAID system is conceptually superior to many contemporary digital systems both in concept and those developed by others. An aspect of the Crackle instruments I find crucial is that control-signal and audio-signal can easily be mixed; a phenomenon that allows the performer to have extremely dynamic control over the behavior of the circuitry, and that is not yet available with small real-time digital systems. But I am 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 Hana's integrated circuits; I do not take that to mean that no confidence in digital instruments will increase as soon as we get the advanced features of analog control conceptualized in a new way inside the hands.

1. TECHNICAL DESCRIPTION

The Hands

The design of this page describes the hardware of the actual prototype. The design of the system was started together with STEER engineer Johan den Bigge, and further development will be pursued with him.

In each hand there are three rows of four keys; these provide pitch control (MIDI key-on, key-off) within one octave. Pitted 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 sensor transmitter aimed at the right hand. The right hand has a matching sensor receiver. The distance between the two hands is measured by comparing the delay of an ultrasonic pulse between the two hands with the original pulse time. The distance 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 stretched. One should note the extreme "irregular" implication of this and, especially, the great amount of control obtained when this key-velocity value is mapped to separate oscillators (X1-operaors) in a FM sound-generation environment.

The left hand provides press-buttons for the selection of three different MIDI channels (1, 2, 3) or all three channels simultaneously. The channel assignment is for both hands. The right hand has press-buttons for program choice (X3: voice 1 to 32). The system is programmed for the prototype, the other down. A reset (to voice "1") has been removed.

An thumb-wheel potentiometer is connected with the pitch bending function on a push-button is available to fragile sustain on and off, used with the XK-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 "play" (because of the octave-"window") or would, unencumbered, have to be played with the two hands' different modes.

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 sensor detector. This is used of the mere interesting possibilities of control with the hands. Continuously controllable timbral changes become possible as a consequence of not only of the changing velocity parameter but also due to interference between the two that at which the "scratch" function generates updates and the attack times of the DTF value. Programming the voice-parameters especially for this "scratch" function one takes a priori for voices nearly rather verti and makes it expressive to the "scratch" control. Sometimes, when sustained is used, extremely dynamic visually, organically connected with its audible appearance. By moving one's arm while having keys pressed one "bends" the sounds.

The circuitry for the sensor-transducers, the pitch bend, the "pull-up" resistors and the model matrix are mounted on the bottom side of the hands. The power is supplied by the analog-to-MIDI converter.

The Image-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 clock, the MIDI-interface and the power supply. There is sufficient RAM on the 6511 to run the programs. There are four 8-bit ports on the 6511, one of which is programmed as a serial port for the MIDI-interfaces, (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 on one of the processors. Program-control was the method since no monitor was available for the single-chip 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 1984, 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 senses the open switches, reads the pitch bend position and measures the sonar distance. These data are mapped through tables to be appropriate sound sequences. It was found that the processor and the program are fast enough to override the data for MIDI-systems. Actually most MIDI systems stall when the "scratch" function is used extensively; MIDI buffer overflow messages were easily obtained on the DRT.

5. CONCLUSION

HANDB

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 "Hands" topics on which

ICM '85 Proceedings 314
research is being done. After that I will describe the concept of the Control Signal Processor which is being evaluated at the moment at TIDZ in collaboration with composer/software-engineer Joni Rys.  

**Hardware**  

The actual (prototype) Hands are strapped with elastic bands to the palms of the performer’s hands. Without being too tight, the bands are fixed with velcro 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 bands that are attached at the wrist so that “free” movements of one’s hands are possible. This will allow a more percussive playing-style.

**Electrode 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 electronic tilt-measuring devices to replace the mercury switches now used for “octave-transpose”.

**Fiber-Optic Sensor**  

We keep a research project by Peter Dessain on the possibilities of using fiber-optic sensors for the detection of finger-, hand- and arm-joint movements. In the Hands these sensors will allow fingers to switch 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 nodes, 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 separately. Keeping track of the channels and available note-programs will definitely increase the need for a visual feedback display.

---

ICMC '85 Proceedings
Program Selection
An extra push-button on the right hand will assign the "senser" value to the program-selection switch. When it is held down, one can uniquely select one of the 32 0-7 voices by moving one slider in and out. "Fine" selection is still possible with the single-step selectors.

Extra Senser Transmitters
Two extra nonal-transmitters will be mounted at the bottom of the hands, aiming downwards. Two matched nonal-receivers, to be clipped to one's trousers, will enable use of phoneme recognition of single hand movements. The resulting values will be assigned by the software to MIDI-control functions. In this way we imagine applications ranging from "modulation-depth" to elaborate "scratching" into extensions of the "interference-control" program.

Pint electronic sensors
Pint electrical accelerometers will be used as data sources for the Senti-algorithm (described later). Pint electric pressure sensors will be added to the "scratching" 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. Pint electric strain and variable resistance paints might be used to detect pressure changes under one's palm, together with the accelerometers and the "scratching pressure sensors that would generate a set of values from which inputs can be made for the Senti-algorithm.

"Crawly" Control
As mentioned before the application of Crawlyouch, Pint pressure can be used. 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
To avoid safety reasons, we opted for wired connection between the hands and the MIDI-controller. It is possible that radio control will be used. If we start building a new hand, sensor requests for familiar verbs have already started to come in.

CYTUGG
The matching, nonal-to-MIDI converter will be extended with an extra button for the storage of MIDI parameter data. This will give experimental re-patching of nonal functions possible with the use of an external terminal ("key"). The memory will be sufficient to store 16 settings of MIDI-command allowing each "patch" key to be individually assigned parameter settings. 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
Some engineers such as Bobo and Paul Spaanman are working on a fast, multi-channel parallel system for the Mac. This will enable us to extend the possibilities of the nonal-to-MIDI control-signal-processing described elsewhere in this report. The output of the converters will be sent to the Macs. Experiments are also being done with 2000 as a host of level 15000 compositional decisions. If it turns out to be successful, the Macs will work together with the "composer" in the "Editor" ( Contr. Design) and, where needed, in assembly language.

Video Synthesizer Control
In the winter of 91, I will perform a piece where two video-camera will scan the movements of the hands during the concert performance. The video signals will be routed to the Fairlight Computer Video/audie for processing and mixing. The resulting images will be projected on two relatively small screens beside the stage. This the hands will, through the MIDI-converters, simultaneously provide control inputs for the video synthesizer and for the sound synthesis instruments.

CV Value Editor
A program is found for the analog-to-dc voltage convertor reconfigures all MIDI controls and generates into a programming device for the CV voltages. The sensor refigures will be the "data-shoulder". The program can be used during the performance by switching a (crawly) function on the hands.

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

This system is designed to increase the quantity of information available from the hands, or from manual controllers in general. The "simple" information generated by finge movements would be applied to a network that would stimulate a complex change in the multiple outputs of this network. The system will be built around wave data tables that are read out by a set of pointers with flexible and manually controlled 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 "crawly" like control system. We expect to have the prototype working in the fall of 1998.

Working with the Control Signal Processor (CSP)
1. A personal computer loaded with a C.E.P. configuration program is connected with the CSP.
2. The 32 waveforms are filled with data.
3. Pointer-moving pathways are defined.
4. The machine is set up.
5. These data are stored on floppy disk as "set-ups" for a piece or a part of a piece.
6. The system is installed in a performance space. A controller is connected 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 waveforms 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 movements 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 table. This will generate a sequence of vectors that, through patching, can be assigned to the control inputs of, for instance, Chant or V230 "generators". The result is a sound sequence. The length of the sound sequence is determined by:
- the algorithms 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 melodic, 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 of pieces.

Proportional finger/hand/foot control of sequence "parameters" initiated by a key depression, is possible by using switches where other parameters, pressure/hand-sensitive keys/heads, accelerometers, etc., are connected with pointer motion functions. Depending on the configuration of the pointer output patching, finger/hand/foot movements can directly control all timbral and color-related aspects of the sound sequence. "Instant", algorithmic control, is possible by using pointer outputs as input for "live" processing 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./Samplitude-system can be turned into a conventional electronic music-keyboard-synthesizer. Filling the wavetables with "intensive vectors" that, through the C.S.P., itself into an autonomous "synthesizer" 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 ways in which I have conceptualised for this task.

The G0Ki (go wild) algorithm detects all control information generated at every moment in the piece. If a certain threshold in the density of control information is passed the algorithm generates new control information by itself and mutates the original control-signals.

The G0N (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 algorithm questions:
- what did I do?
- what did I want?
- was I good?
- what do I do now?

The T.E. algorithm (That's 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 G0Ki 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 G0Ki algorithm. This will be performed as a suggestion.

The Effect algorithm is the one that has my full attention at the moment. It is possible to build any sort of any controller nowadays. As long as 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 (quasimodulation of control-data by the control signal processor).

Figure 2

CONTROL - SIGNAL - PROCESSOR

ICMC '95 Proceedings
creation of raw 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 NANDs. The NANDs provide a vast amount of control over the musical material, but this is far from a guarantee of a good piece. I see live composing as the richest musical experience. The demands on the composer/performer 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 coming 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 simultaneous listening/placing. I mean hearing the developing structure of a piece (a piece not composed in advance in the comfort of the erudite composer's 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.