The Electrumpet, a hybrid electro-acoustic instrument

Hans Leeuw

Utrecht school of the arts
Department Music Technology
Postbus 2471
1200 CL Hilversum (NL)
hans.leeuw@kmt.hku.nl

Abstract

The Electrumpet is an enhancement of a normal trumpet with a variety of electronic sensors and buttons. It is a new hybrid instrument that facilitates simultaneous acoustic and electronic playing. The normal playing skills of a trumpet player apply to the new instrument. The placing of the buttons and sensors is not a hindrance to acoustic use of the instrument and they are conveniently located. The device can be easily attached to and detached from a normal Bb-trumpet. The device has a wireless connection with the computer through Bluetooth-serial (Arduino).

Audio and data processing in the computer is effected by three separate instances of MAX/MSP connected through OSC (controller data) and Soundflower (sound data).

The current prototype consists of 7 analogue sensors (4 valve-like potentiometers, 2 pressure sensors, 1 "Ribbon" controller) and 9 digital switches. An LCD screen that is controlled by a separate Arduino (mini) is attached to the trumpet and displays the current controller settings that are sent through a serial connection.

Keywords: Trumpet, multiple Arduinos, Bluetooth, LCD, low latency, OSC, MAX/MSP.

1. Introduction

Quite a number of papers have been written on music instrument augmentation for NIME conferences. However, looking at papers written on brass instrument augmentation the number is quite low. This is especially true when looking for extensive articles describing the design process, the type of sensors used and the possible locations these sensors can be placed which is more commonly described for other instruments [1]. Furthermore this article has been written from the perspective of an instrumentalist designer.

However, a number of people did work on the trumpet. Ben Neil has made the Mutan trumpet [2]. Thomas Craig and Bradley Factor have made the Trumpet MIDI controller [3], Sukandar Kartadinata has made and helped with three different custom trumpet controller extensions for Jonathan Impett [4,5] Axel Dörner and Rajesh Mehta [6]. Other examples are the EVI and MDT [7].

The examples above either involved rebuilding the trumpet [2], or took the trumpet purely to play different sounds (e.g. saxophone / guitar) [7]. The trumpets designed by Sukandar [4,5,6] are more closely related to the Electrumpet but these were custom- built and an extensive article describing the design process and choices involved has not yet been published.

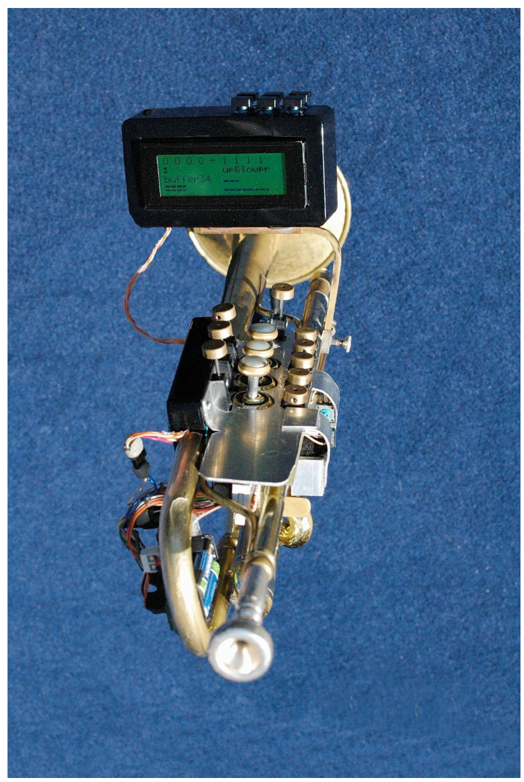


Figure 1. The Electrumpet

2. Design considerations

The most widely used setup in live-electronics with acoustic instruments is that of a laptop, a separate controller, the instrument itself and a microphone.

Imbalance between the performer's instrument and controller command, a variety of focal points (instrument, controller, screen) and the lack of a direct connection between controller action and musical action can all contribute to the confusion of an audience.

In the light of the discussion above the choice has been made to develop an instrument/setup that integrates the expressiveness of the acoustic instrument in the design of an electronic instrument.

So the sensors of the Electrumpet are placed directly on the instrument of the performer, which is nothing new in itself [3 – 10], the fact that it relates so directly to the normal handling of the acoustic instrument while still being a separate controller is a new element however.
Another important aspect of the Electrumpet in this light is the direct responsiveness of the instrument (low latency). This has been achieved by good timing of the Bluetooth serial connection, compacting of the data and the use of multiple controllers (two Arduinos) and program instances (three MAX/MSP instances).
Last but not least, the controller is wireless, which vastly improves the player's freedom of movement and removes visually distracting wires.

2.1 Expression

Facial and body expressions of the performer are important in the perception of music:
"At a perceptual level, they signal important melodic, harmonic, and rhythmic events. Facial expressions may reflect the additional concentration that is needed to perform notes or passages that are unexpected or tonally unstable. Performers may also intentionally introduce facial expressions and other gestures as a way of sharing with listeners their understanding of the musical significance of such events. .... In this way, visual aspects of performance signal that performers are not merely producers of sound but are themselves listeners, highlighting the musical activity as a shared experience between performers and listeners." [8]
One remark on this connotation may be that when the connection between controller and sound is too obvious the experience of 'hearing what you see' easily becomes 'cheesy' and 'shallow'.
One of the beauties of acoustic music is hearing and seeing the mastery of a skilled instrumentalist in controlling an instrument that has inherent chaotic behaviour (sound wise). This translates into a design in which the controls and the software are constructed in such a way that we can see and hear the connection between sound and big gestures while at the same time these gestures should not become too obvious and straightforward but do require fine motor skills.

2.2 Instrument or piece?

In the article "Principles for Designing Computer Music Controllers" [9} Perry Cook states about the Cook/Morrill trumpet [10] that trumpet players have 'spare bandwidth' which they can use for triggering electronics for example. Then he argues that '*the live recording and triggering of looped notes was 'a miserable failure'* and makes the choice to revert to *'simple, nearly stateless interactions' in combination with fairly autonomous compositional algorithms.*
The statement on 'spare bandwidth' has also been proven true for the Electrumpet. Keeping track of loops of sounds and controlling them is possible as well but it depends on the straightforwardness of their application whether or not it makes musical sense.
The experience of using the Electrumpet in a Big Band as a sound controller for looped sounds while simultaneously playing the trumpet has already proven to be quite satisfactory.
The ultimate goal is to use live-recorded sound in a rhythmical and controllable way in conjunction with rhythmical acoustic play.
Integrated compositions with live-electronics in which the electronic sounds can be played with the same (kind of) articulation, diction and control as the acoustic instrument should be possible. The integration of acoustic and electronic controls influencing the same originally acoustic sound source seems to be a possibility as well.
The future objectives are twofold:
1) Playing the Electrumpet in improvised and composed pieces as a normal instrument.
2) Pieces especially written for the Electrumpet.

3. Designing the Electrumpet

The Electrumpet can be a DIY device making use of the advances of different recent developments in technology. Microcontrollers in IO devices have become affordable and have been assembled in easy-to-use, compact and robust prints that can be used for prototyping purposes. Wireless techniques are at a level where they can be used almost without distortion and with low latency. There are many ready-made components and fine hobby tools on the market that are affordable.
It is good to stress the importance of the advance of open source hardware here. Until recently we were either bound to ready-made solutions (expensive and inflexible) or building from scratch. The complete construction of the Electrumpet took an inexperienced builder only two months.

3.1 Electrumpet Setup

The whole setup of the Electrumpet is compact and it still fits easily in a normal trumpet bag.
Sound is picked up with a microphone and transferred into digital data using a sound card (FA101, Edirol). The data is sent to the computer through a firewire connection and subsequently processed in two software programs (two patches programmed in two instances of MAX/MSP).

The parameters controlling the behaviour of the software patch are generated by a separate controlling program that facilitates the Bluetooth connection and translates the data coming in from the Arduino.

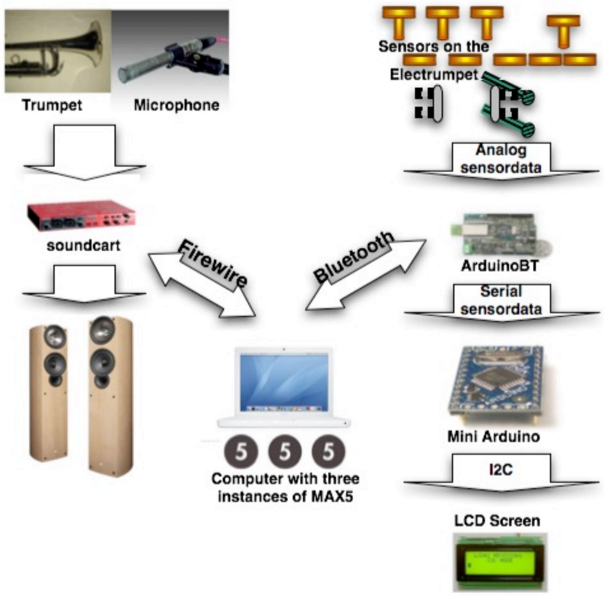


Figure 2. Electrumpet Setup chart

4. Designing the hardware

4.1 Designing the hardware, deliverables:

Quite a number of mechanical and operational issues had to be addressed when designing the instrument. The following list shows the issues and solutions.

4.1.1 Low latency and wireless.

In order to function as a true musical instrument the latency had to be less than 20 ms and preferably 10 ms. This fact combined with the requirement of a wireless interface led to the choice for an Arduino Bluetooth board. The latency is currently down to about 15 ms.

4.1.2 Alignment of buttons and potentiometers parallel to the normal valves of the trumpet.

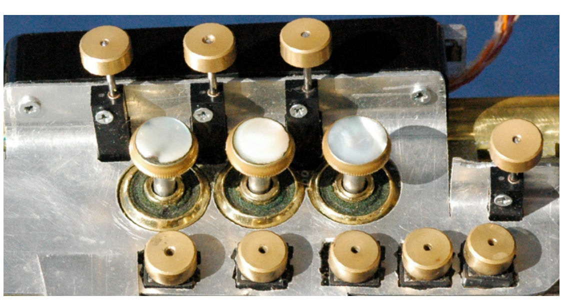


Figure 3. Potentiometers and Buttons with normal valves

With this alignment the trumpet player does not have to move his hand while playing the electronic parts, the thumb of the right hand being traditionally a point of reference.

4.1.3 Potentiometers with the same mechanical response as normal trumpet valves.

Potentiometers were used which have similar mechanical responses as trumpet valves. The valves' travelling distance is also to the same order (1.2 cm).

4.1.4 Easily mountable and removable (attachment to the trumpet).

The entire controller is clipped to the instrument or held in position by existing trumpet parts.

4.1.5 Feedback.

When selecting presets it is important to know the configuration on the Electrumpet since not everything is always audible. An LCD screen provides this feedback.

4.1.6 No influence on the acoustic trumpet sound.

This requirement is met. The attachment of the LCD screen (a marching band music stand) had to be replaced when the first placement influenced the sound.

4.2 Designing the hardware, other possibilities

The above features were the starting point of the design but further control opportunities remained.

4.2.1 Pressure sensors

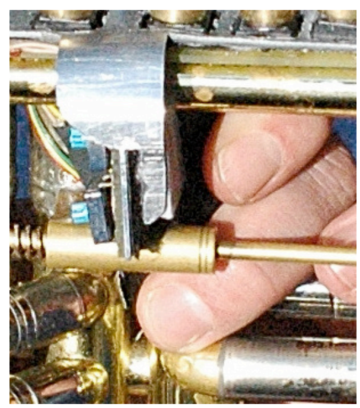


Figure 4. Using the Pressure sensors

The fingers of the left hand that hold the valve housing can be used to control sensors since they are not used for playing the trumpet but only for holding it. There is no possibility to move the fingers but it is quite possible to apply pressure. Pressure sensors placed on the third valve for the index finger and the middle finger are easy to manipulate and can be used as a controller.

4.2.2 Slide buttons

A trumpet player uses the thumb of the right hand as a counter force when pushing the trumpet valves. This is the same for the digital valves and 'action' buttons. There is some room for side movement in the thumb joint. Experimentation led to an arrangement of four different buttons that can be pushed with this thumb.

4.2.3 'Ribbon' controller

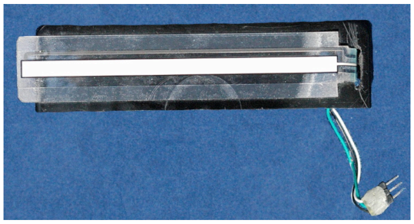


Figure 5. Ribbon Controller on plastic frame

There is a ribbon controller on the left side of the Electrumpet which can be played with the left thumb.

4.3 Assembling the sensors, switches and Arduino board into one design:

The main circuit board is used as a solid base for the construction. Five buttons are attached directly to the board and the four other buttons are also paired on small strips of circuit board.
The circuit board itself is held in position by the top part of the valves. In this way the circuit board does not hinder the trumpet player's handling of the normal trumpet and it can easily be removed. On top of the circuit board sits an aluminium plate that has been shaped in such a way that the sensors and buttons which are not directly mounted on the circuit board can be attached to it. These are the digital valves and the four slide buttons.
The two pressure sensors are glued onto a PVC cylinder halve that is clipped on the third trumpet valve.

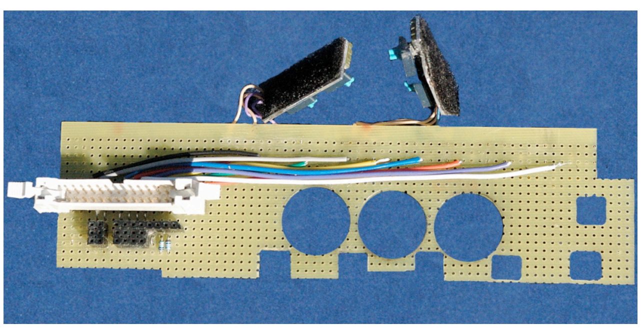


Figure 6. The underside of the circuit board

The area under the circuit board is relatively safe from accidental touching. Here all connectors to the off-board sensors are placed as well as the main connector to the Arduino board section. The Arduino board and the batteries are mounted on a vacuum-formed plastic frame. The frame clips onto the trumpet. The Arduino and the circuit board are connected by means of a flat cable.

5. Computer interfacing:

The Arduino captures all sensor data. There are a few considerations when programming the Arduino:

5.1 Timing / sampling rate / latency / connectivity

Low latency is important for timed musical actions. To achieve minimum latency the Arduino only sends data after one start command.
This type of connection seems to be conductive to stability as well. It is possible to freely move from the computer (Mac book) up to a distance of 10m without losing the connection.
The data is packed by bitshifting in as few bytes as possible. The sampling rate of the measurements is brought up to 500 Hz. this is also an aid in bringing down the latency.
Currently the latency is typical 15 ms, measured by comparing the sound coming from a switch that is being pushed with the change in value corresponding to that same switch. Occasionally the latency goes up to 30 ms however.
By using three instances of Max/MSP it is possible to optimize for controller data processing (no sound processing, short event and scheduler intervals), sound processing (long sound vector sizes and long event and scheduler intervals) or sound timing (short sound vector sizes, short event and scheduler intervals). The three instances are connected through Soundflower (for sound) and OSC (for controller values).

5.2 Data processing for use on both the computer and the LCD

The most efficient way of sending the data is to send them raw, but it has to be ensured that the interpretation of the data on the LCD screen is the same as on the computer. Therefore some pre-processing on the Arduino board was required. This also facilitated the need to pack the data as dense as possible.
All analogue sensor data is packed as single bytes (losing 2 bits of possible Arduino resolution). Toggles and scroll switches are packed together in bytes by bitshifting.
The slider buttons are also processed on the Arduino and sent as bytes to make sure that the same slider setting is sent to the computer as well as to the LCD.
The LCD has its own controller (an Arduino mini). The control of the LCD could have gone through the Arduino Bluetooth as well but this would have slowed down that controller and it is very helpful for the overview to have a separate capture and display of the data.

6. Interaction with the instrument

6.1 Functionality:

The placing of the sensors and buttons on the instrument has been designed with future virtuoso playing in mind. The interaction is described in the following list:

6.1.1 *Electronic valves*[1]*:*

With a normal trumpet there are actually two ways of using the valves. Pushing down a valve means lowering the tone by one, two or three semitones. The rhythm of the notes is determined by pushing down valve(s) (combinations) with some tongue action. The valves can also be used to alter the upper structure of the notes ("halve valve").

To map the trumpet player's skills in the device two possibilities are halve valve techniques to alter the quality of the sound or pressed down valve combinations for timed forms of electronics.

6.1.2 *Pressure sensors*[1]*:*

The pressure sensors are mounted on the third valve of the trumpet. By squeezing the index and middle finger of the left hand the value of the parameters that are connected to these sensors can be changed. Holding this to a particular value is not easy. The pressure sensors are especially useful for parameters the player wants to change while using the right hand on the valves.

The pressure sensors are topped with rubbery tape to have more haptic feedback.

6.1.3 *'Action' buttons:*

These are buttons that are mounted directly next to the trumpet valves on top of the lead pipe of the trumpet. Three of these are also aligned exactly with the trumpet valves. Since they are easy accessible they are especially useful for timed use.

6.1.4 *'Slide' buttons*[1]*:*

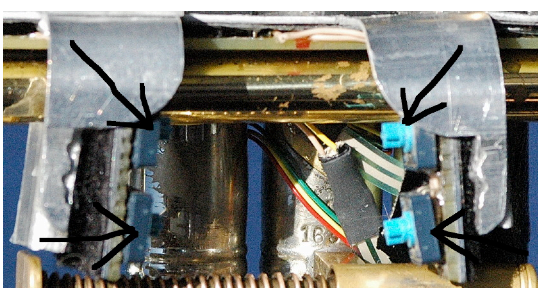


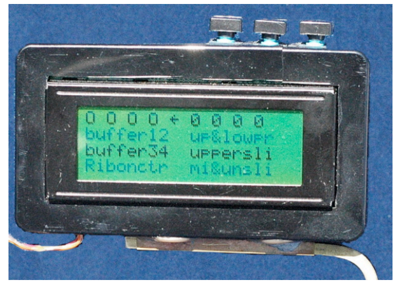
Figure 7. Side view on slide buttons

These could also be called 'scroll' buttons. The configuring of the buttons is such in the software on the Arduino that it is possible to slide three parameter values. The top two buttons slide one parameter, the bottom two buttons another and together they can change a third. These buttons respond slower than the digital valves.

6.1.5 *"Ribbon' controller*[1]*:*

The 'Ribbon' controller is the only continuous controller that can jump in values. The controller is played with the left thumb. It is possible to scratch or thumb it like a slap bass or use it simply as a controller with jump capability. This is especially useful when driving preset vectors with this controller.

6.1.6 *LCD Screen:*



Full advantage of the wireless capability of the Electrumpet can only be taken if there is the possibility of feedback on the instrument itself. The LCD screen lets the player walk around freely, looking at the trumpet instead of a computer screen and reading control values and sheet music at the same time.

7. Using the data

7.1 Gestures and control:

The Electrumpet makes use of the trumpet player's skills. To also map the fine motor skills the first and second order derivative of the analogue sensor data have to be mapped to parameters as well.

The combination of valves has musical meaning for a trumpet player (pitch when pushed completely, sound when pushed partially). The mapping of combined data from different valves should be explored as well.

7.2 DATA collection, analysis and OSC-commands

Currently the data is captured in a standalone application (built in MAX/MSP) that also analyses part of it. The control over all parameters (both direct data and the data that is the result of the analysis) is the aim as a player. The current work is on simple analysis of valve movements and mapping to rhythmical / continuous play. The use of the 'action' buttons in different modes is also part of this exploration.

The idea is to have the standalone application send useful information as OSC-commands that can easily be applied to the control of sound data in other applications.

8. First experiences

Having used the instrument for almost a year in solo and ensemble settings it becomes more and more obvious that more effort is devoted to sorting out the precise scaling of the controllers than would be the case with a 'classical' fader- or rotary- controller. There is a different process going on: a trumpet player wants the instrument to respond in a certain way; a response that is similar to the instrument without the controller. That feeling can be achieved by precise scaling of the controller values, by logical dividing the different parameters over the sensors

[1] outputs digital to a value between 0 and 255

and by choosing and programming parameters with an 'instrumental feel' in mind.

When using the Electrumpet as an effect controller, the advantage over, for example, a foot controller is the speed at which settings can be changed.

Capturing sounds in (fft) buffers and manipulating these with the sensors is more typical. The digital valves are used to trigger envelopes over the processed sound. Since this is done in a low latency environment any processed sound can be played in a rhythmical way independent of the acoustic rhythm.

There is still no sensation of playing an instrument that is both acoustic and electronic but that is probably because all available time has been spent on designing rather than on practicing...

9. Future work

9.1 Designing the hardware, future enhancements:

One of the key actions of the trumpet player is the use of lip tension and breathing control to produce sound. That capability is not yet implemented in the instrument as a means of control but will be added in the future.

For timed playing the 'action buttons' can be used. In the next version buttons for changing presets, entering edit mode and loading a new patch will be implemented.

9.2 Software control and gestures

As described in 7.1 the analysis of first and second order sensor data plus the analysis of sensor data combinations and the mapping of these to the control of audio data will make this instrument attractive to trumpet players. Further in-depth exploration of this feature will be part of future research.

9.3 Use of notation for effect control

Musical (note) notation can actually address (the timing of) electronic effects since the trumpet player is used to translate notes in valve actions and the valve (and button) positions are the same on the Electrumpet.

10. Final Words

The Electrumpet is a valuable contribution to the existing family of augmented instruments. Specific features are: The low latency Bluetooth connection, the use of an onboard LCD, the combination of two Arduinos for specific tasks on the device and the use of three instances of MAX/MSP next to each other. These are all applications of recent developments in (open source) hard and software. These new combinations proved to be stable. Hopefully this device will inspire other people to augment their instrument in a similar manner.

The equal integration of the electronic instrument with its acoustical counterpart has to be worked out further but has already proven its potential. More (technical) details can be found at www.electrumpet.nl

11. Acknowledgments

With respect to electronics:
Jorgen Brinkman (STEIM), Gerard van den Boomen Faculty Industrial Design of the TU Eindhoven (Tue),
With respect to manufacturing:
Bert Lonsain, Wim Scheepers (TUe), Atelier Pfeiffer.
With respect to software:
Adrian Freed, Ali Momeni, Michael Zbyszynski. (CNMAT), Emmanuel Jourdan (IRCAM), Arjan Loogman (HKU)
Gwion Lopez for making pictures.
With respect to funding:
The design was sponsored by the Netherlands 'Fonds voor de Scheppende Toonkunst'.
Furthermore I would like to thank the following people for a variety of reasons:
Jeanine Kierkels, Colette van Montfort, Meike van den Broek, Rutger Menges, Jan Belon, Menno Meijer (TUe), Jorrit Tamminga, Marcel Wierckx (HKU), Jos Zwanenburg (Conservatory of Amsterdam), Cort Lippe (University at Buffalo), Edwin van der Heide, Michel Waisvisz (STEIM), Ann La Berge.
Finally I want to thank all developers of the Arduino environment and contributors of the Arduino Forum. I think they are doing a great job in democratizing the world of innovation and any way make it a lot more fun!!!

References

[1] A. Freed, D. Wessel, M. Zbyszynski "Augmenting the Cello" in *Proc. of the 2006 International Conference on New Interfaces for Musical Expression (NIME06), Paris, France . pp 409 - 413*

[2] B. Neill "The Mutan Trumpet" [http://www.benneill.com]

[3] T. Craig and B. Factor "The Trumpet MIDI Controller" [http://courses.cit.cornell.edu/ee476/FinalProjects/s2008/twc22_bef23/twc22_bef23/index.html]

[4] Jonathan Impett, A Meta-Trumpet(er). In: *Proceedings of the International Computer Music Conference (ICMC) in Århus, Denmark, 1994. pp 147 – 150*

[5] Jonathan Impett, Projection and Interactivity of Musical Structures in Mirror-Rite. In: *Organised Sound, vol. 1 no. 3, 1996, pp. 203 Ð 211*

[6] Sukandar Kartadinata "The Gluiph: a Nucleus for Integrated Instruments" in *Proc. of the 2003 International Conference on New Interfaces for Musical Expression, Montreal, Canada. pp 180 - 183*

[7] S. Marshal and J. Morrison "Morrison digital trumpet" [http://www.digitaltrumpet.com.au/]

[8] W. F. Thompson, P. Graham, F. A. Russo (2005). "Seeing music performance: Visual influences on perception and experience" in *Semiotica 156, pp 203 - 227*

[9] P. Cook, "Principles for Designing Computer Music Controllers" in *Proc. of the 2001 International Conference on New Interfaces for Musical Expression, Seattle, USA.*

[10] D. Morrill and P. Cook, "Hardware, Software, and Compositional Tools for a Real-Time Improvised Solo Trumpet Work," in *Proc. Of the International Computer Music Conference, Columbus, OH, 1989. pp 211 - 214*